Global-local Knowledge Coupling Approach to Support Airframe Structural Design

Global-local Knowledge Coupling Approach to Support Airframe Structural Design

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 23 december 2014 om 10.00 uur

door

Haiqiang WANG
Bachelor of Mechanical Engineering
Nanjing University of Aeronautics and Astronautics, People's Republic of China
geboren te Lu'an, Anhui province, People's Republic of China.

Dit proefschrift is goedgekeurd door de promotoren:
Prof.dr.ir. M.J.L. van Tooren

Copromotor: Dr.ir. G. La Rocca

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof.dr.ir. M.J.L. van Tooren	Technische Universiteit Delft, promotor
Dr.ir. G. La Rocca	Technische Universiteit Delft, copromotor
Prof.Dr.ir. L.L.M. Veldhuis	Technische Universiteit Delft
Prof.dr.ir. R.Benedictus	Technische Universiteit Delft
Prof.Dr. T.Tomiyama	Cranfield University
Prof.Dr. W. Liao	Nanjing University of Science and Technology
Ir. C. van Hengel	Fokker Aerostructures

ISBN 978-94-6203-740-3

To my parents and my wife

Contents

Acknowledgements v
Summary vii
List of Symbols and Acronyms xi
Chapter 1. Introduction 1

1.1 Airframe as a complex system 2
1.2 The aircraft design and development process 3
1.3 Research problem description 4
1.4 Research objective 5
1.5 Thesis outline 6

Chapter 2. Airframe design process 9

2.1 Related theory 9

2.1.1 Function-behavior-structure framework 9
2.1.2 Systems engineering 11
2.1.3 Multidisciplinary design optimization 14
2.1.4 Concurrent Engineering 16

2.2 Airframe structural design process 18

2.2.1 Global design process 19
2.2.2 Local design process 20
2.2.3 Global knowledge and local knowledge 22

2.3 Concluding remarks 23

Chapter 3. Proposed approach 25

3.1 Global-local knowledge coupling approach 25
3.2 Knowledge based engineering 27
3.3 Design and Engineering Engine 29
3.4 Demonstration systems 30

3.4.1 Fuselage ADEE 32
3.4.2 Fuselage panel ADEE 34
3.4.3 Movable ADEE 35

3.5 Automated FE-based structural analysis 37

Chapter 4. Implementation and verification of the fuselage ADEE 45

4.1 Overview of the fuselage ADEE....45
4.2 Methods used for discipline analysis....48

4.2.1 Load calculation method....48
4.2.2 FEA-based sizing method....50

4.3 Implementation of the fuselage-structures class....54

4.3.1 HLPs of the fuselage structural members....56
4.3.2 CMs for load calculation....61
4.3.3 CMs for FEA-based sizing....63

4.4 Application cases and discussion....66

4.4.1 Conventional aircraft fuselages....66
4.4.2 Prandtl plane fuselage....70
4.4.3 Results and discussion....71

4.5 Concluding remarks....73

Chapter 5. Implementation and verification of the fuselage panel ADEE....75

5.1 Overview of the fuselage panel ADEE....75
5.2 Methods for discipline analysis....77

5.2.1 Structural analysis method....77
5.2.2 Cost estimation method....79
5.2.3 Weight evaluation method....80

5.3 Implementation of the panel MMG....80

5.3.1 Panel HLPs....80
5.3.2 CMs for structural analysis....86
5.3.3 CMs for cost estimation....91

5.4 Demonstration of the fuselage panel ADEE....92
5.5 Discussion....94

Chapter 6. Implementation and verification of the movable ADEE....97

6.1 Overview of the movable ADEE....97
6.2 FEA-based weight estimation method....100
6.3 Implementation of the movable MMG....103

6.3.1 Movable HLPs....104
6.3.2 CMs for weight estimation....113

6.3.3 CMs for cost estimation 115

6.4 Application case and discussion 116

6.4.1 Description of the application case 116
6.4.2 Design of experiment and response surfaces 118
6.4.3 Optimization results and discussions 119

6.5 Concluding remarks 121

Chapter 7. Conclusions and future work 123

7.1 Conclusions 123
7.2 Limitations and Future work 124

Appendix A The PCL-writer 127

A.1 Three types of PCL files for automated FEM generation 127
A.2 Different numbering methods between GDL and PATRAN 127
A.3 LISP functions of the PCL-writer 128
A.4 Use of the PCL-writer 133

Appendix B Input parameters for the fuselage MMG 135
Appendix C Validation of the fuselage panel sizing module (Zee type stringer) 139
Appendix D Empirical equations for fuselage weight estimation 143
Appendix E Geometry of lifting surfaces of PraP300 145
Appendix F DoE results for the fuselage ADEE 147
Appendix G Input parameters for the panel MMG 149
Appendix H Input parameters for the movable MMG 153
Appendix I Movable ADEE DoE results 157
References 159
Samenvatting 169
About the Author 173

Acknowledgements

Firstly, I would like to express my appreciation to the supervisors who made this research possible. I am greatly indebted to Prof. Michel van Tooren and Dr. Gianfranco La Rocca who accepted me to do my PhD study and have supported me throughout. Michel: first of all, thank you for your patience and your trust in me, especially when I felt frustrated in the middle of my PhD study. Secondly, thank you for offering me the internship at Fokker Aerostructures where I got to know how aircrafting works in the real world. Finally, thank you for checking my draft thesis in the weekends when you were supposed to be enjoying life with your family. Gianfranco: thank you for opening the door to KBE for me. Our daily discussions about KBE were very helpful to me, helping me to define my research objectives and to achieve them. Your sense of humor glues all the KBE guys together, and your efforts to correct this thesis are also highly appreciated, any remaining mistakes are all my own.

I am very grateful for the financial support provided by the Chinese Scholarship Council. Special thanks are also due to Prof. Liao Wenhe who was my supervisor during my graduate studies in China. He supported me my application for the scholarship, and continues to support me.

My thanks also go to Miranda and Claire for helping me to improve my academic writing.

Thanks to my officemates, Maarten, Dipanjay, Marcel, Durk and later on Lex, Yan and Hamid. It is very nice to continue the tradition of at least three coffee-breaks per day. Thanks for the excellent intellectual atmosphere provided by my fellow colleagues in our group: Ali, Tobie, Eric, Li, Xiaojia, Feijia, Reiner, Maurice, Fengnian and Peijun. I would like to thank Irene for organizing the regular TAPAS meetings that gave me the opportunity to exchange ideas with my fellow TAPAS colleagues, Huajie, Xiaojia, Wim, Beatriz and Maria, from welding to damage tolerance, health monitoring and costs.

I would like to thank Prof. Egbert Torenbeek, Mark and Roelof for sharing their knowledge of aircraft design. Special thanks go to Prof. Torenbeek for introducing me to the Dutch culture, Dutch paintings, organ music, as an international student having a guide was invaluable. I would also like to extend my gratitude to Arvind for being an enjoyable neighbor.

I am so lucky to have both a Dutch and a Chinese knowledge base. Thanks are due to the engineers from Fokker Aerostructures for their willingness to share their knowledge with me, a non-Dutchman: Ton, Ed, Max, Frank van Dalen and Frank Hodes, thank you. Ke Liang, from the Aerospace Structures group and Li from the same group as me who swam with me in the same think tank brainstorming about finite element analysis and aircraft design. Thanks for always being there when I needed to consult.

I would also like to thank the people who supported me when dealing with administrative duties, Lin, who helped me a lot before my admission, and during the time when I had just arrived in the Netherlands, Nana and Bettie. Thanks are extended to Michiel for helping me solve those computer problems.

There is always life besides work. Thanks to my housemates, Yan and Ping, and former housemate Xiaoyu, Li, Wei for so much fun. In 2013 we had two special guests (Li's parents) over for a visit, Uncle Mo who introduced me to the guitar and Aunt Mo who made a delicious dinner every evening. Thanks for making our apartment a real home. I would also like to thank Huajie and Lu Zhang, Lu Wang and Shuzheng, Hao and Xinhui, Xuefei and Xun, Haoliang and Qiwen, Dan and Zhengpei for providing me with good examples of loving couples. Thanks are also given to Yongjia, Ke Tao, Zhijie,

and Xinyuan for the parties at which we could comment on anything we liked. Thanks are also extended to my Chinese friends from other departments at the Faculty of Aerospace Engineering, who I will not mention individually, but you know who you are.

Last but never least, I would like to thank my family. A special thanks to my parents and younger sister whose love encouraged and continues to encourage me to try the chocolates of life. Papa and Mama; I feel so proud and lucky to be your son. Thanks for always encouraging me to do what I want to. And to my loving wife, Qi, thanks for your willingness to share the magic chocolates with me. I look forward so much to continuing the magic with you by my side.

Summary

The outsourcing that has taken place in the aircraft industry over the last few decades has created a globalized supply chain from and to a limited number of original equipment manufacturers (OEMs). This has led to multi-level design due to the shift from airframe subsystem design to suppliers. Increasingly OEMs focus on requirement allocation and definition of airframe subsystems and verification at a global level, whereas suppliers focus on the realization and improvement of airframe subsystems at the local level.

Relying on a supply chain for innovative designs and builds can cause OEMs to have insufficient bottom-up knowledge about subsystem design, in particular, the innovative local designs, e.g. composites and new production methods, however, in the overall aircraft conceptual design phase, the analysis and evaluation of different subsystem designs, by OEM internally, rely heavily on assumptions and estimations which are usually based on statistical/empirical data. Although global designs can be quickly analyzed using assumptions and estimates, this risks costly design changes if the assumptions and estimations are proven incorrect in the later overall aircraft design phases. Suppliers who have detail-level knowledge should be involved early in the overall aircraft conceptual design phase, creating various local designs, and conducting more accurate analyses and evaluations of these designs. Early local design studies can help suppliers help OEMs to reduce the risk of design changes related to incorrect assumptions and estimations, and convince OEMs of the benefits of new material and new production methods.

The objective of this research was to develop a design approach which can support **suppliers** to perform local design fast from which critical results, i.e. cost and weight, can be generated during the overall aircraft conceptual design phase. A fast airframe subsystem design is highly beneficial for suppliers wishing to increase their competiveness, providing fast response and being flexible in the overall aircraft conceptual design phase. It is also beneficial for OEMs to reduce the risk of design changes due to incorrect assumptions and estimations.

Several issues in the current design process that hamper a fast study of airframe subsystems were identified in this research, some of which have to be addressed from the supplier's side.

1) The dependency of suppliers on the OEMs to get coherent, consistent and timely design information, e.g. geometry and load cases, needed to start local design. This dependency causes suppliers wait until all the required information is available from the OEMs in the overall aircraft preliminary design phase. Therefore, the suppliers cannot proactively participate in the overall aircraft conceptual design, in which the airframe subsystem design relies heavily on assumptions and estimations.
2) The manual processes used by suppliers to update computer aided design (CAD) and analysis models to follow design changes at the global and local level. In the overall aircraft conceptual design phase, both the global and local design are not fixed yet and tend to change. Manually model updating at local design level takes significant engineering efforts, and hence slows down the supplier's response to the changes in the global design.
3) There is a lack of multidisciplinary design optimization (MDO) capability and capacity at a local design level due to this lack of MDO knowledge and a lack of tools to build parametric product and process models. Therefore, in the short conceptual design phase, suppliers often just deliver a (few) feasible design solution(s) instead of a family of Pareto design solutions.

To address these issues, and hence to increase supplier competitiveness, a **global-local knowledge coupling** approach is proposed, which comprises two modules at the global and local design level. The module at global design level is the cross-over, which is used as a substitute for global design and provides the inputs required for starting a local design. The cross-over is used to make the global and local designs concurrent in the early aircraft design phase. The module at the local design level is a set of parametric product and process models of airframe subsystems used to automate repetitive design actions at local design level, such that the analysis and evaluation of subsystem designs can be quickly performed. Knowledge based engineering (KBE) is adopted to implement the two modules for two main purposes: 1) parameterization of product models that allows automatic model (re)generation; 2) automation of pre-processing to prepare inputs for disciplinary analysis tools. Multidisciplinary design optimization is used as the technical implementation mean of the proposed approach to automate the process of finding an optimal design for a complex airframe subsystem.

Three demonstration systems are developed, each of them formed as a design framework, called the Airframe Design and Engineering Engine (ADEE), which is a specialized **Design and Engineering Engine (DEE)**. The design and engineering engine (Tooren, 2003) is a MDO system aimed at supporting and accelerating the design process of complex products, through the automation of non-creative and repetitive design activities. The verification design systems are the fuselage ADEE, the fuselage panel ADEE and the movable ADEE.

One of the main contributions of this research is to identify the issues in the airframe design process which involves OEM and suppliers, and how these issues can be solved for quickly performing local design in the aircraft conceptual design phase. Another contribution lies in the development of the global-local knowledge coupling approach and its demonstration systems for the new design approach, which provide tools and methods to address these issues. Each verification tool is an ADEE, which is supported by KBE to perform global design and local design in an automatic fashion, such that cross-over can quickly generate the required inputs for local design and the local design module can quickly generate and analyse various subsystem design variants.

The fuselage ADEE is used to address issue 1 by increasing design independence for panel suppliers

The fuselage ADEE is implemented as a cross-over, in which finite element analysis (FEA) based weight estimation is developed to capture the effects of material and structural layout on fuselage weight. The global knowledge is captured in the cross-over, including the knowledge of how to generate fuselage outer mould line (OML) and knowledge of how to perform disciplinary analysis such as load calculation and structural analysis using FEA. The ADEE is validated using data from fuselages of conventional aircraft such as the ATR 42, Fokker 100, Boeing 737-200, Airbus A320-200 and Airbus A300B2. The fuselage ADEE is also used to estimate fuselage weight of a joint wing aircraft.

The fuselage panel ADEE is used to address issue 2 by automating repetitive model (re)generation for local design

The fuselage panel ADEE is the local design module of the global-local knowledge coupling, which comprises a parametric panel product model and disciplinary analysis models, i.e. structural analysis, cost estimation and weight evaluation models. The fuselage ADEE is a cross-over which provides inputs for the fuselage panel ADEE. A KBE-enabled parametric panel product model is implemented in the fuselage panel ADEE to model various configurations of fuselage panels flexibly, which are

composed of skin with multiple layers and back-up structural members, such as frames and stringers. These structural members are modeled based on the OML generated by the fuselage ADEE. The structural analysis uses global-local FEA in which a global FE model is obtained from the cross-over to predict the overall fuselage behavior, whereas a refined FE panel model is built for investigating panel behavior. The local panel process knowledge is captured in the panel ADEE so as to automate the panel modeling, structural analysis, parametric bottom-up cost estimation and weight evaluation. Using the accelerated local panel design process, the local panel design can quickly respond to the change of global design, while the model consistency between global and local levels can be guaranteed.

The movable ADEE is used to address issue 3 by automating repetitive design actions in the MDO process

The movable ADEE is developed to perform cost/weight multi-objective optimization of movable structures, e.g. rudders and elevators, including large topology variations of the structural configuration. The KBE-enabled modelling module of this ADEE is able to model very different product configurations and variants and extract all data required to feed the weight and cost estimation modules, in a fully automated fashion. The weight estimation method uses FEA to calculate the internal stresses of the structural elements and an analytical composite plate sizing method to determine their minimum required thicknesses. The manufacturing cost estimation module was developed on the basis of a cost model available in the literature. The capability of the framework is successfully demonstrated by designing and optimizing the composite structure of a business jet rudder. The study case indicates that this ADEE is able to find the Pareto optimal set for minimum structural weight and manufacturing cost quickly.

The demonstration systems developed demonstrate that the global-local knowledge coupling approach can support suppliers wishing to perform fast airframe subsystem design in the overall aircraft conceptual design phase.

List of Symbols and Acronyms

ACARE	Advisory Council for Aeronautics Research in Europe
ADEE	airframe design and engineering engine
AI	artificial intelligence
CAD	computer-aided design
CAE	computer-aided engineering
CDE	computer design engine
CM	capability module
CLIOS	complex, large-scale, interconnected, open, and sociotechnical systems
COTS	commercial off-the-shelf
FBS	function behavior structure
FEA	finite element analysis
FEM	finite element model
FIDO	framework for interdisciplinary design optimization
GDL	general-purpose declarative language
GLARE	glass laminate aluminum reinforced epoxy
GUI	graphical user interface
HLP	high level primitive
IGES	initial graphic exchange specification
KBE	knowledge based engineering
KBS	knowledge based systems
LE	leading edge
LISP	LISt processing
MDCAD	multi-disciplinary concept assessment and design
MDO	multi-disciplinary design optimization
MMG	multi-model generator
MOB	multidisciplinary optimisation of a blended wing body
MOKA	methodology (and software tools) oriented to knowledge based engineering applications
NASA	National Aeronautics and Space Administration
NASTRAN	NASA STRuctural ANalysis
NGA	non geometric attribute
NVA	non value added
OEM	original equipment manufacturer
OML	outer mould line

PCL	PATRAN Command Language
PPS	polyphenylene sulfide
PrADO	preliminary aircraft design and optimization
RFP	request for proposal
SE	systems engineering
STEP	standard for the exchange of product model data
TE	trailing edge
UML	unified modeling language
WB	wing box

Chapter 1. Introduction

In the first 100 years of flight many aircraft configurations were developed, and the technical evolution of large civil transport aircraft has resulted in the current dominant aircraft configuration: a cylindrical fuselage, a pair of cantilever wings and a vertical and horizontal tail. The dominance of this configuration, leading to the decomposition of the airframe shown in Figure 1.1 has caused large scale rationalization of the aircraft industry and in today's industry only a handful of original equipment manufacturers (OEMs) remain. These OEMs focus on aircraft development and integration. Most of the subsystems and components are made in extensive supply chains. An example of the supply chain of a Boeing 787 is shown in Figure 1.2.

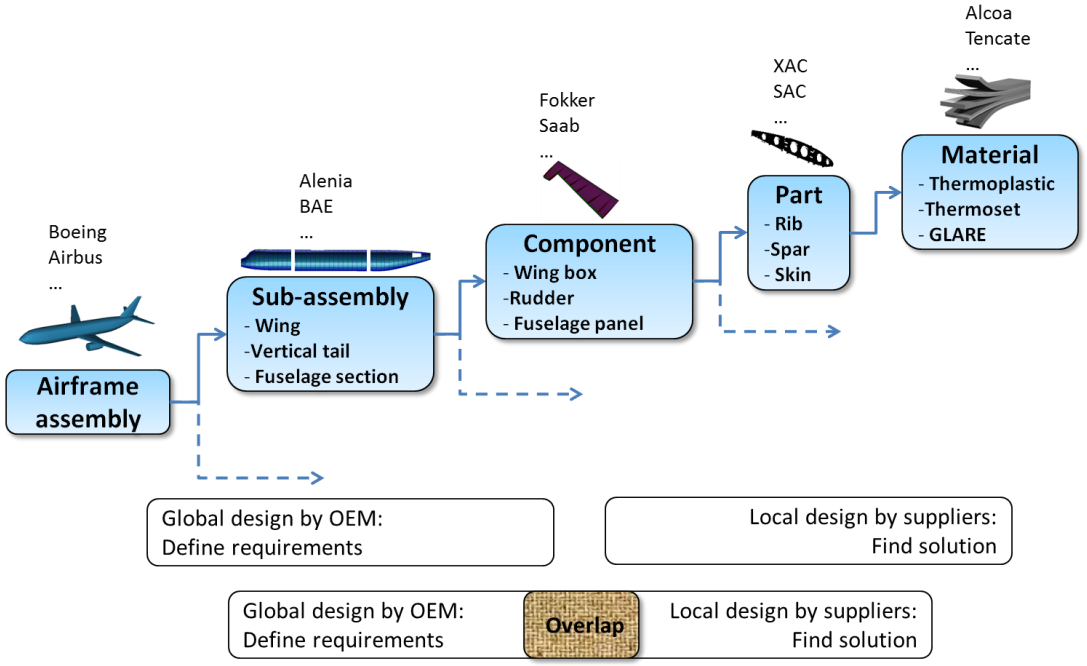


Figure 1.1: Airframe decomposition (suppliers in this thesis are Tier 1 suppliers who design and built the sub-assemblies or components)

Although the industrial setting of a limited number of OEMs with a well-developed supply chain is effective for risk sharing and recurring cost reductions, it easily hampers innovation and continuous development. The role of an OEM is to provide suppliers with requirement specifications, while the role of aircraft suppliers is to design and build airframe subsystems that comply with the requirement specification. This division, however, is far from strict: suppliers should participate in discovering requirements and OEMs should be involved in the solution finding and elaboration. This function overlap between OEMs and suppliers is very important during early product development. Part of the overlap can be implemented using a design approach proposed in this research to allow suppliers to perform early design studies for airframe subsystems, and to participate in requirement discovery.

Two terms are defined here for clarity:

global design: a design at the system level. In this research, the global design is the overall aircraft design. The term "global" refers to the system level.

local design: a design at the subsystem/component level. In this research, the local design is an airframe sub-assembly/component design. The term "local" refers to the subsystem/component level.

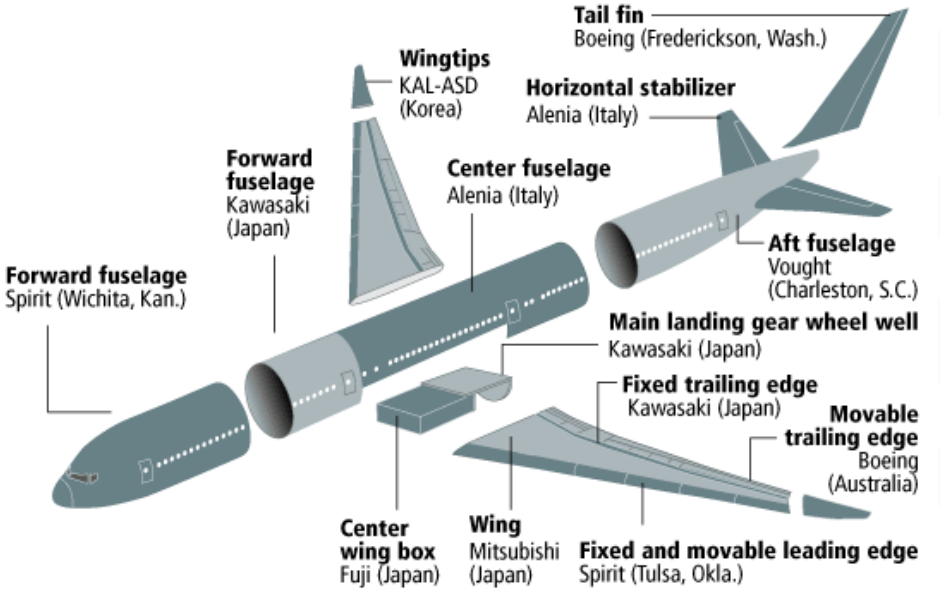


Figure 1.2: An example of the supply chain of a Boeing 787 (courtesy of Boeing)

1.1 Airframe as a complex system

The term complex system is rather broadly defined. Various definitions for complex system can be found depending on the field they are derived from (Sussman et al., 2000; Suh, 2005; Miller et al., 2007; Mitchell et al., 2009). Sussman's (2000) definition is:

Complex systems are systems composed of a group of interrelated components and subsystems, for which the degree and nature of the relationships between them is imperfectly known, with varying directionality, magnitude and time-scales of interactions.

Four types of complexities are defined by Sussman (2000) in complex, large-scale, interconnected, open, and sociotechnical systems (CLIOS):

- structural complexity: exists when a systems consists of a large number of interconnected parts
- behavior complexity: exists when the output or behavior of a system is difficult to predict
- nested complexity: exists when interactions of a system within another system are difficult to quantify
- evaluative complexity: exists when different stakeholders view different aspects of system performance in different ways

An airframe is, by nature, a complex system and Sussman's four types of complexities can be observed in airframe design. Structural complexity is apparent because of the complex airframe product breakdown and interrelations between the breakdown structures. For example, an Airbus A380 wing consists of fuel tanks, engine support, control systems, 10 aluminum alloy skin panels, 62

ribs, 3 spars, 157 wing stiffeners, 22 control surfaces and 375,000 fasteners such as nuts, bolts, and rivets (Minnett and Taylor, 2008).

Behavior complexity surfaces when the behavior of an airframe system has to be determined. The different airframe designs, such as different structural concepts, materials and production methods, lead to different airframe behaviors, i.e. structural performance, cost and weight. The impact of these design choices on airframe behaviors is difficult to quantify.

Nested complexity can be observed in the airframe supplier chain where the OEM works at the global level and suppliers at the local level. The airframe subsystem design process is nested in the overall aircraft design process, see Section 1.2.

Evaluative complexity surfaces as different stakeholders in airframe design have different views on the optimal design. Very often, OEMs prefer the minimum weight design as the optimal design while suppliers see the minimum cost design as the optimal design.

Apart from the four types of complexity defined by Sussman, airframe systems also feature modelling complexity, that is the complexity associated with the need for coherent, consistent and comprehensible multi-view models at the detail design level. Modelling complexity surfaces in the repetitive model adaption and preprocessing for multiple views. This complexity can be observed in the airframe design process discussed in Chapter 2.

1.2 The aircraft design and development process

The airframe subsystem design process is nested within the overall aircraft design process, as shown in Figure 1.3. According to classical aircraft design textbooks (Anderson, 1999; Torenbeek, 1982), the overall aircraft design process can be divided into three phases, namely the conceptual, preliminary and detail design phases. The division of the design process is not only an academic argument, but appears in the project development process in OEM's daily practice. The Airbus milestone model in which three design phases ranging from M2-M4, M3-M5 and M5-M7 can be distinguished is shown in Figure 1.4 (Pardessus et al., 2004). Between M2 and M3, the OEM identifies the most promising concept and it optimizes the concept at aircraft level between M3 and M4. In the aircraft conceptual design phase, a lack of design information and the complexities of an airframe system make it difficult for an OEM to analyze and evaluate different subsystem designs.

During the aircraft preliminary design phase, the airframe subsystem design process starts after an OEM issues a tender that includes a list of requirements for airframe subsystem design. The conceptual design of an airframe subsystem is performed by several suppliers, and feasible design options, the subsystem design and building principle, which satisfy all the requirements are delivered to the OEM. The OEM receives the design options, evaluates them, selects a design which best meets the requirements and decides which supplier wins the contract. After receiving the request for a proposal from an OEM, the conceptual design phase of subsystems starts, as shown in Figure 1.3. Once all the subsystem designs are finished, the final selection of suppliers is made. After that, the winner of the tender process receives more requirements from the OEM, and the detailed design of the airframe subsystem starts when more data/information becomes available. The airframe subsystem design process involves engineers from different domains and this process is discussed in more detail in Section 2.2.

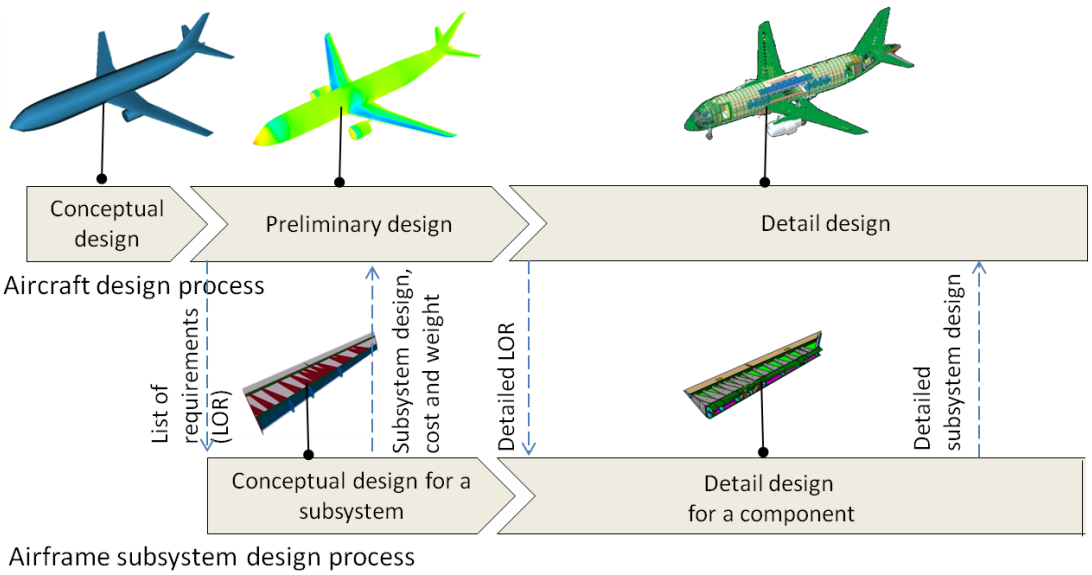


Figure 1.3: Airframe subsystem design process, concurrent with the aircraft design process

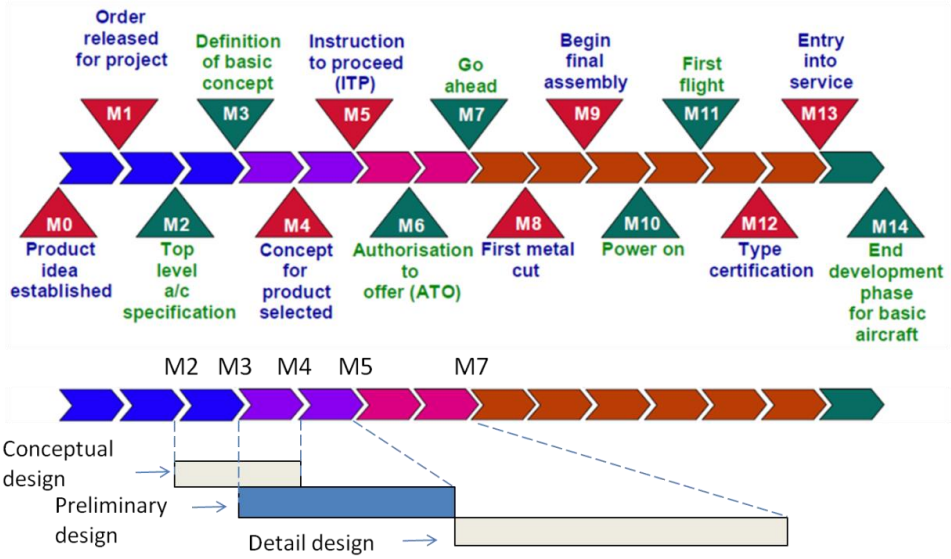


Figure 1.4: Airbus milestone model, courtesy of Airbus[1]

1.3 Research problem description

Looking at the airbus milestone model, it can be seen that the aircraft design process is a top-down process. As progressive milestones occur, an increasing number of design details are added, which then allows for a more thorough analysis and sub-level exploration (Cooper, 2010). In the early design of a complex airframe system, an OEM works at the global level to allocate overall performance requirements and general geometric constraints to the subsystems, whereas suppliers at the local level utilize a detail-level knowledge about the airframe subsystems to create subsystem designs which

[1] In the Airbus process, there is an overlap between the conceptual design phase and the preliminary design phase, which is the phase ranging from M3 to M4.

achieve the performance requirements passed from the global level while meeting the geometric constraints.

Relying on the supply chain for innovative design and build can get an OEM into the situation that it has insufficient bottom-up knowledge about subsystem design, in particular, the innovative local designs, e.g. composites and new production methods, however, in the overall aircraft conceptual design phase, the analysis and evaluation of different subsystem designs, by OEM internally, heavily rely on assumptions and estimations which are usually based on statistical/empirical data. One example of such an estimation is the Class II weight estimation method, the component weight estimation method, (Torenbeek, 1982) which relates subsystem weight to several geometric parameters and performance parameters using semi-empirical equations. Although a global design can be quickly analyzed using assumptions and estimates, this practice risks costly design changes if the assumptions and estimations are proven incorrect in the later overall aircraft design phases. Nevertheless, suppliers who have detail-level knowledge can be involved early on in the overall aircraft conceptual design phase by creating various local designs, and by conducting more accurate analysis and evaluation of these designs. With the early local design study, suppliers can help the OEM to reduce the risk of design changes related to incorrect assumptions and estimations, and a local design study is also beneficial for suppliers to gain competiveness by participation in the design process.

The early local design should be quickly performed because of the short lead time of the overall aircraft conceptual design. The current local design speed is slow, mainly due to the repetitive design actions that take place in the local design process. An example of repetitive design actions is the repetitive design actions caused using an increasing number of physics-based analysis tools. Compared with empirical data and formulas, physics-based analysis tools can help engineers to understand the causality of novel designs better. Using the physics-based analysis tools, the engineers can gain more confidence by observing the behavior of novel designs in the early design phases, however, a lot of time is consumed in generating various design variants, and pre-processing and post-processing the models for these physics-based analysis tools, especially when the overall aircraft design is not fixed in the conceptual phase, and a change of design is often necessary.

The slow local design pace leads to limited subsystem design space exploration. Suppliers are not able to perform a multidisciplinary design optimization studies or to provide a set of Pareto optimal designs, and instead only one or a few feasible designs are delivered.

1.4 Research objective

The research objective for this research was:

To develop a design approach which can support suppliers to quickly perform airframe local design from which critical results, i.e. cost and weight, can be generated in the overall aircraft conceptual design phase.

The design approach is called global-local knowledge coupling in this thesis. As the name indicates, the approach integrates global knowledge, design knowledge at the global level, and local knowledge, design knowledge at the local level. Global knowledge is applied to provide the inputs needed to start a local design, whereas local knowledge is captured to accelerate the local design process by automating the repetitive design actions.

Knowledge based engineering (KBE) was adopted to enable the development of the approach for two main purposes: 1) parameterization of product models that allows automatic model (re)generation; 2) automation of pre-processing to prepare inputs for disciplinary analysis tools. The proposed approach and KBE is introduced in Chapter 3.

Using the critical results from an early and fast local design, the knowledge required for the global design is expected to increase in the overall aircraft conceptual design phase, as shown in Figure 1.5.

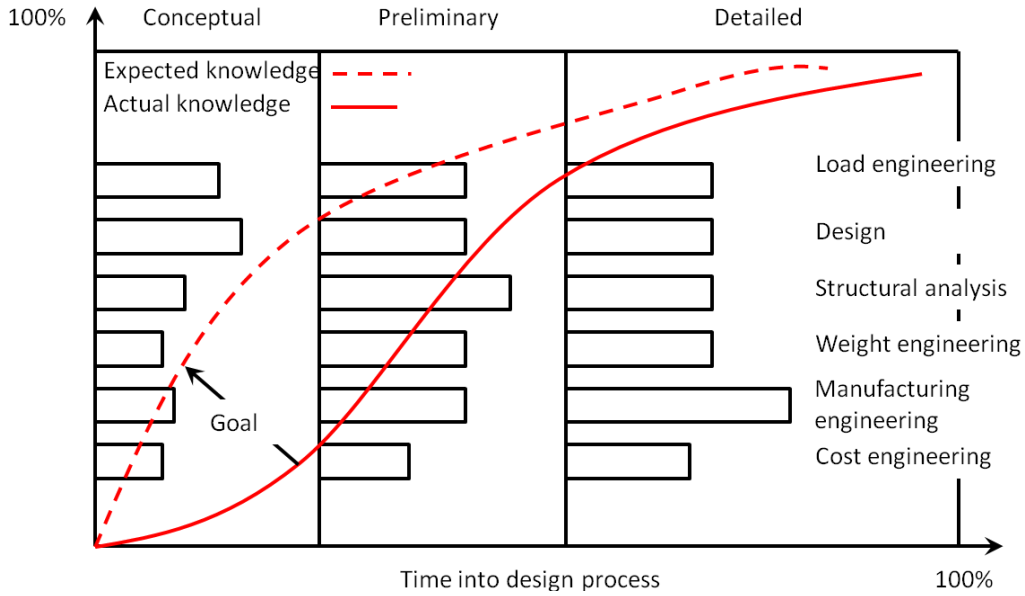


Figure 1.5: Knowledge increase in the overall aircraft conceptual design phase (modified from Schrage et al., 1991)

Although the design approach proposed in this research can be applied to all airframe subsystems, only movables and fuselage panels of transport aircraft were used to illustrate this approach as example subsystems. The verification of the proposed approach was done by building several design systems, referred to airframe design and engineering engines (ADEEs). These are the fuselage ADEE, the fuselage panel ADEE and the movable ADEE. The ADEEs are specialized design and engineering engines (DEE) (La Rocca and Tooren, 2003) which are used to support multidisciplinary airframe structural design. The DEE concept is introduced in Chapter 3.

Some statements are made about the goal of the research referred in this thesis for the purpose of clarity. This research was NOT

- the design and development of the complete tools/methods to perform the airframe design
- done to validate completely the results with the real structures. Although it would be ideal to compare the design results with a real design, data about detail designs was very difficult to access for the research.

1.5 Thesis outline

The global-local knowledge coupling approach proposed in this thesis is aimed to supporting suppliers in the task of quickly performing an early design for complex airframe subsystems during the overall aircraft conceptual design phase. The approach was verified using three prototype systems, and these

systems were further verified using application cases. The detailed design process of the airframe structural design, involving OEM and suppliers, and the issues in the current design process that will be addressed by the proposed approach are identified and discussed in Chapter 2. The proposed approach is discussed in Chapter 3. The framework of the prototype design systems that can embody the approach is discussed in Chapter 3. The related techniques/methods are discussed, and the requirements for demonstration systems are given in Chapter 3. Following these requirements, the implementation and verification details of the demonstration systems are described in Chapter 4-6. Finally, conclusions and suggestions for future work are given in Chapter 7.

Chapter 2. Airframe design process

The detailed airframe design process and the related research are discussed in this chapter. The related research: function-behavior-structure (FBS) framework, concurrent engineering (CE), systems engineering (SE) and multidisciplinary design optimization (MDO) are discussed in Section 2.1. The current airframe design process in conceptual and preliminary design phases is described in Section 2.2. The stakeholders, i.e. engineers from OEM and suppliers, and the design actions for which they are responsible are described in detail. Finally, the issues that hamper fast local design are identified in Section 2.3.

2.1 Related theory

The function-behavior-structure (FBS) framework is first described in Section 2.1.1 to provide a scientific basis for modeling the process of designing and the iterative nature of a generic design process is made explicit. The airframe design process is discussed within the context of the FBS framework.

Systems Engineering (SE) is discussed in Section 2.1.2. SE is often used by engineers to provide guidance for system development. In SE system design is viewed as a top-down process, however, the top-down approach should in some way be linked with a bottom-up approach, which can capture the quantitative impacts of subsystem designs, e.g. cost saving and weight saving, on system performance to help OEMs to make correct decisions about the overall aircraft design in the first place.

Design of complex airframe structures involves several disciplines, such as cost estimation, weight estimation and structural analysis. Discussed in Section 2.1.3, multidisciplinary design optimization (MDO), as a quantitative side of SE, is used to automate the search for a more balanced system optimum, instead of a mono discipline optimum. In addition, MDO can be used to find not just one feasible design solution, but a set of optimal designs.

CE is discussed in Section 2.1.4 to address the need in the aircraft industry to minimize lead time by making processes parallel. The two processes that are made parallel here are the conceptual phase of the overall aircraft design, i.e. the global design, and the subsystem design. To achieve this concurrency, airframe subsystem suppliers should be able to assume the inputs needed to start the local design, instead of relying on the inputs provided by the global design.

2.1.1 Function-behavior-structure framework

In the design research literature (Gero et al. 1990; Tomiyama et al. 1990) one can find many references to the function-behavior-structure (FBS) framework which is used as a scientific basis to model a design process as a set of distinct design activities. This section seeks to understand the design process in a scientific manner from the point of view of the FBS framework.

The FBS framework formally defines that the relation between a function and a structure fulfilling the function can only be quantified at the behavior level. Function (F) is the purpose of the design object, whereas structure (S) is the product components and their compositional relationships (Gero and Kannengiesser, 2004). Behavior (B) is the measurable attributes or effects that can be derived from its S. According to the FBS framework, B is specialized into expected behavior (Be), the

"desired" behavior, and behavior derived from structure (Bs), the "actual" behavior. Among the three variables, F, B and S, S is the only one on which designers can make a direct decision. In addition, two further notions are introduced to the FBS framework: requirements (R) which represent intentions from the client that come from outside the designer, and a description (D) that represents a depiction of the design created by the designer, shown in Figure 2.1.

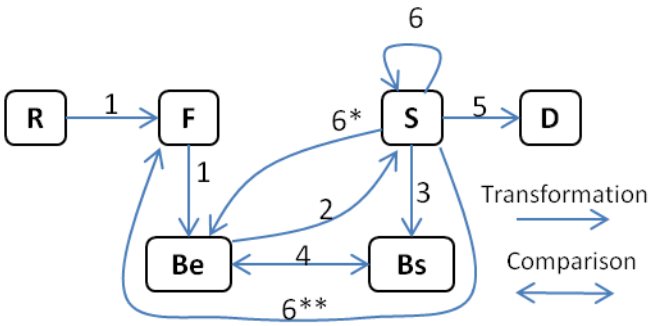


Figure 2.1: FBS framework (Gero and Kannengiesser, 2004)

In the FBS framework process has six steps that are considered to be fundamental for designing.

1. Formulation (R→F → Be)
2. Synthesis (Be → S)
3. Analysis (S → Bs)
4. Evaluation (Be ←→ Bs)
5. Documentation (S → D)
6. Reformulation (S → S*/Be*/F*)

The design cycle is drawn in Figure 2.2 according to the FBS framework to show the iterative nature of a design process. A similar iterative design process, shown on the right side of Figure 2.2, is suggested by van Tooren (2003).

In the case of airframe structural design, F is extensively expressed in specific load sets and safety requirements regarding residual strength in the case of partial failure. Customer requirements, such as requirements on cost and weight, can be transformed into F in terms of constraints. Over the first 100 years of flight, F has led to a set of well defined standard requirements from EASA and FAA.

The S denotes the airframe product breakdown and the interrelations between the breakdown structures. A definition of S must be given, such as the dimensions of all the structural members of an airframe, the material used for each structural member, the structural layout which describes how those members are geometrically related, the production method used for each structural member and the assembly concept which describes how to assemble those members.

According to the FBS framework, S can only be quantified at B, which are strength, stiffness and cost and weight. As discussed in Chapter 1, airframes are characterized by structural complexity, which is the complexity associated with product breakdown structures and the interrelations. To reduce this complexity, both global design and local design are involved separately in the design cycle that transforms F to S. First, Be of the entire airframe is predicted in the formulation step of the global design. Then, the physical architecture of the entire airframe is defined at the global level, where the airframe is decomposed into several subsystems. After that, the load sets, cost budget, weight budget

and geometric constraints are allocated as F for each airframe subsystem. The S of airframe subsystems is left to be determined during the local design. The design cycle mentioned above is followed to determine S of each airframe subsystem at the local level. After S of all the airframe subsystems is determined, Bs of the entire airframe is analyzed and compared with Be of the entire airframe to determine whether another iteration of the design cycle at the global level is required.

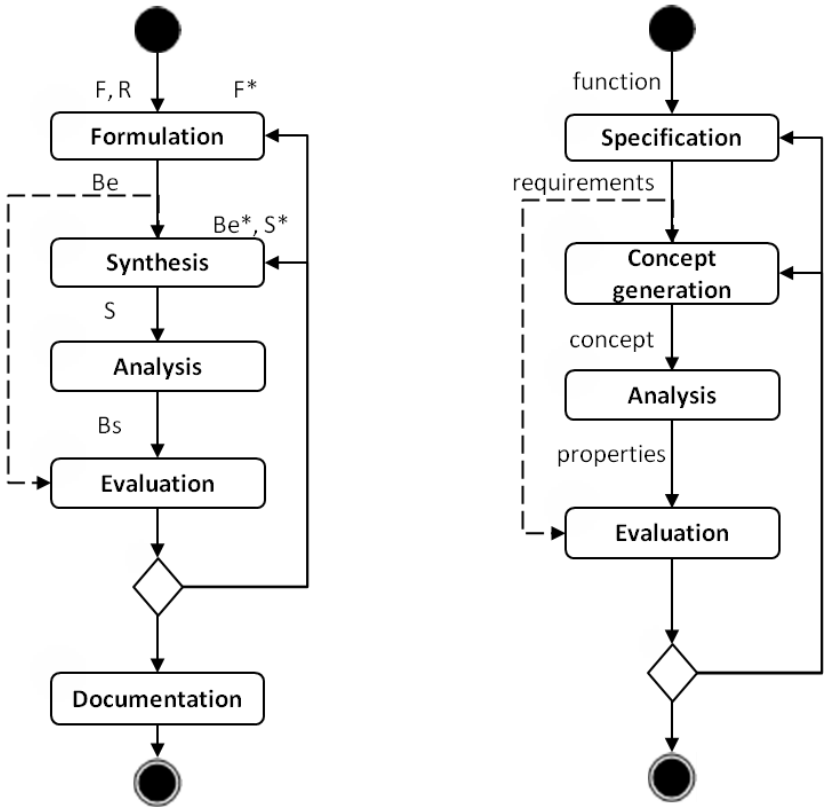


Figure 2.2: Design cycle according to the FBS framework, (left); design cycle suggested by van Tooren (2003) (right).

2.1.2 Systems engineering

Systems engineering (SE) has been widely used to provide guidance during the formulation of the design process for a complex system such as an airframe structure. The scope of SE compasses the entire life cycle of a system (USDoD, 2001). There are several definitions for SE used by international societies and organizations, i.e. NASA, IEEE, INCOSE, USDoD, etc. The SE definition from INCOSE (2006) is given below.

"An interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation…"

The SE process is often seen as a ***top-down***, comprehensive, interdisciplinary and iterative problem solving process (USDoD, 2001; Blanchard and Fabrycky, 2011; Cooper, 2011). The sequential phases of the product development is shown in Figure 2.3. When the design phase of, in our case, an airframe progresses from left to right as diagrammed in Figure 2.3, decisions are made concerning more and

more design details, which then allows for more thorough analysis and sub-level exploration of the airframe.

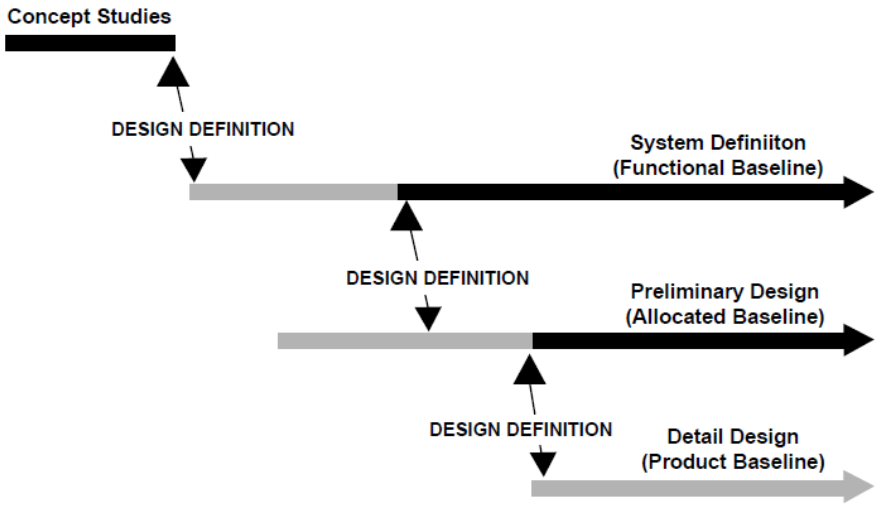


Figure 2.3: Design phasing used in systems engineering (DoD, 2000)

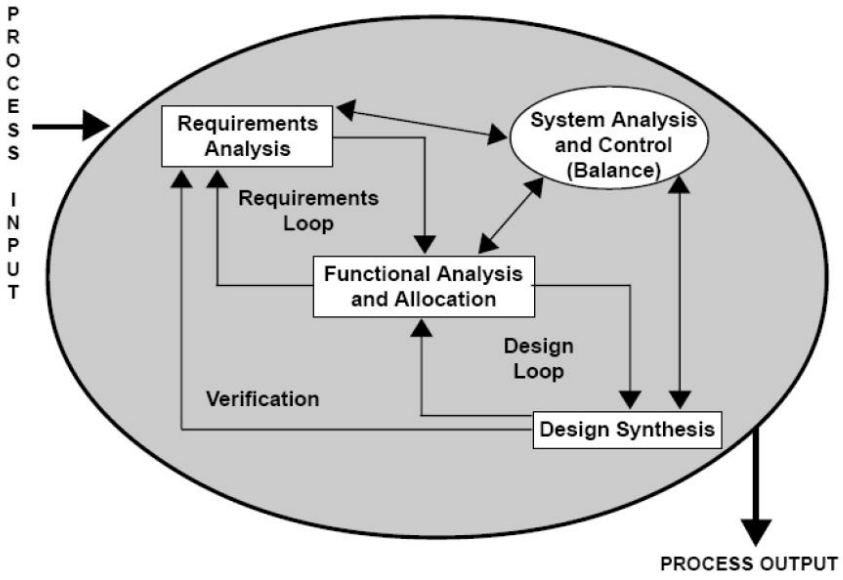


Figure 2.4: System engineering process (USDoD, 2001)

The each design phase in Figure 2.3 is by nature iterative. The US Department of Defense (DoD) Acquisition University (2001) has an overview of the SE process (Figure 2.3), which combines several fundamental activities: requirements analysis, functional analysis and allocation, and design synthesis, balanced by system analysis and control. The SE process defined by the Defense Acquisition University can be used to illustrate the inner working of each design phase shown in Figure 2.3.

The "Vee" process model from INCOSE (Figure 2.5) is a SE process model often used in practice, which views the system development process as a design and verification process. The left side of the "Vee" shows the system top-down design process, in which the system is decomposed into

subsystems, then the subsystems are further decomposed into components – a large system is broken into smaller and smaller pieces through many levels of decomposition. As the system is decomposed, these requirements are decomposed into more specific requirements, which are allocated to subsystems. Then, the subsystem design starts with a set of specific requirements. After the design of components is finished, verification is performed by testing from the subsystem level to the full system level.

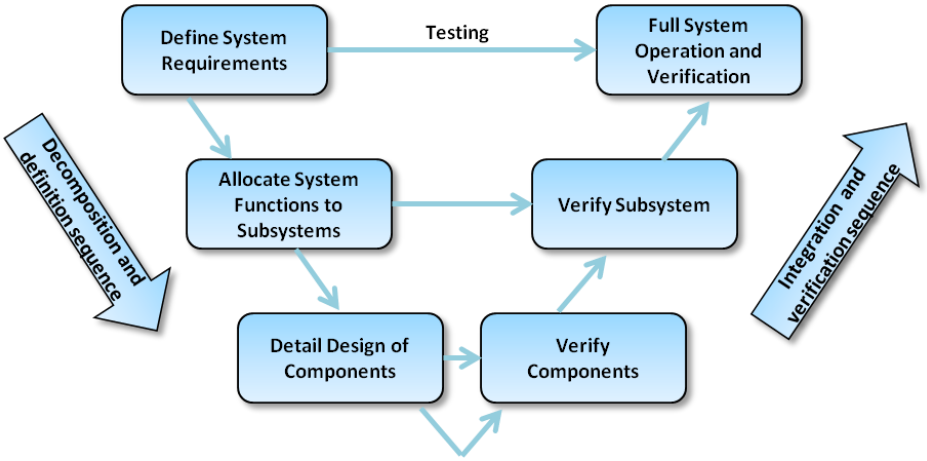


Figure 2.5: "Vee" process model. Adapted from Forsberg and Mooz, 1992

All the SE process models discussed above view system design as a top-down process. As mentioned in Chapter 1, the Airbus milestone model supports the idea that the aircraft design process is an example of the top-down process. Blanchard and Fabracky (2011) has stated that this top-down approach should be married with a ***bottom-up*** approach for product realization, as illustrated in block 5 of the system engineering morphology shown in Figure 2.6. This combined approach is adopted to support the design of complex aircraft wings, spontaneously taking into account knowledge from downstream, e.g. cost estimation and manufacturing rules, in the top-down wing design process (Cooper, 2011).

In the aircraft industry supply chain, the top-down bottom-up approach is important. OEMs utilize global knowledge to allocate functions for subsystems at the global level, whereas suppliers uses local knowledge to create the design and to provide the assurance necessary to the OEMs to levy the top-down requirements in the first place (Cooper, 2011). As new material and manufacturing methods are increasingly applied in airframes, the OEMs have to assess new technologies in the early design phase to investigate their potential for weight and cost saving to enhance the performance of an entire aircraft. The weight and cost saving potential of a new technology should not be quantified only at the local level, relying only on statistical/empirical data, but the physics-based methods, i.e. bottom-up parametric cost estimation and FEA, should also be used. Therefore, the local design should be involved in all early design phase of the global design to obtain the right estimations for cost and weight at an early stage in the aircraft design process.

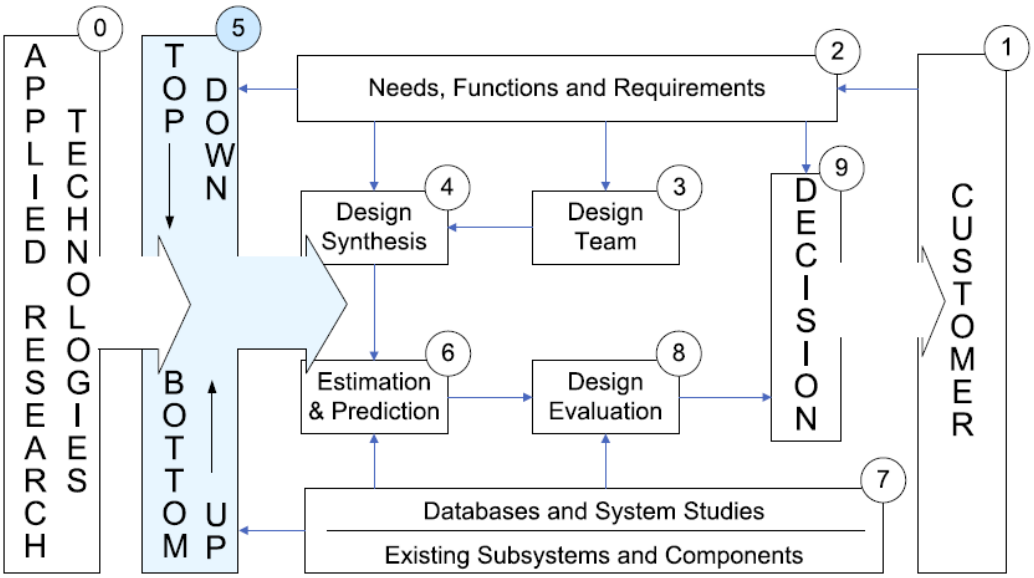


Figure 2.6: System engineering morphology for product realization (Blanchard and Fabracky, 2011). Adapted from Cooper (2011)

Van Hinte and van Tooren (2008) point out that SE is not a recipe for a good design per say: SE can be seen as a qualitative framework of tools that can be used to solve ill-posed problems (van Tooren and La Rocca, 2008). The designing of an airframe system is a complex and multidisciplinary process, which requires both qualitative and quantitative skills and tools: MDO, as a quantitative side of SE, provides a framework of tools to automate the search for an optimum design of a complex system.

2.1.3 Multidisciplinary design optimization

Multidisciplinary design optimization (MDO) is used to automate the process of finding an optimal design for a complex system (Simpson et al., 2011). In a case where the design objectives of the individual disciplines conflict, MDO can be used to make a good compromise between these disciplines by integrating all of them in the design and optimization process. Hence, a more balanced system optimum can be found, instead of finding a mono discipline optimum.

Extensive research has been conducted since MDO was first proposed in the 1980s.Most of the MDO literature is focused on a mathematical approach to formulate the optimization problem formally, e.g. definition of objective function, design variables and constraints, and the organizational strategy, e.g. decomposition and coordination strategy, (Vanderplaats et al., 1984; Sobieszczanski-Sobieski et al., 1997; Kroo et al., 1997; Alexandrov et al., 1997; Allison et al., 2006; Tosserams et al., 2010). In these studies, MDO is based on the assumption of availability of all the required engineering tools, e.g. discipline analysis tools and optimization tools, a problem space, e.g. system requirements, and a solution space, e.g. system concept solutions.

To implement the quantitative possibilities of MDO, a MDO framework is necessary to provide a computational environment to support the MDO study. A generic MDO framework comprises three functional modules (Vandenbrande et al., 2006).

1) A modeling and analysis module which is able to estimate the performance of multiple aspects for each design variant.
2) A design explorer module which generates design points to sample the design space conveniently, hence it defines the design variants that are indicated by the variable vectors.
3) An optimizer to spot the most promising area in the design space, based on the feedback responses. Optimizers are often used to perform both function 3 and function 2.

Three different implementations of MDO frameworks can commonly be found according to the types of modeling implementation used, namely geometry-less, grid-perturbation and geometry-in-the-loop implementation (Vandenbrande et al., 2006).

In the context of an airframe MDO study, the optimization should simultaneously consider different disciplines, in this case mainly cost, weight and structural analysis. The objectives are normally expressed as cost and weight. The design variables are usually selected from the parameters which determine the detail characteristics of an airframe structure, such as material type used and the dimensions of all the structural members, the manufacture method and the structural layout. The design constraints are mainly distinguished into structural constraints, e.g. strength, stability and stiffness requirements, and manufacturing constraints, e.g. minimum thickness step.

Several MDO frameworks have been found in literature which can be used to address the cost/weight multi-objective optimization problem. Kassapoglou (1997) optimizes a stiffened panel for minimum weight and minimum cost. The optimal panel design differs with different defined optimization objectives. Curran et al. (2006) optimizes a fuselage stiffened panel with the DOC as the optimization objective which is formulated as a combined function of the manufacturing cost and the structural weight. The structural sizing in the studies by Kassapoglou and Curran is based on closed-form analytical formulas for common failure modes, such as flexural buckling, local buckling inter-rivet buckling and material failure based on the material allowable stress. The cost estimation is calculated using semi-empirical equations which link cost with the design variables considered in the structural sizing.

More complex airframe structures cannot be sized using the structural sizing method in the design framework proposed by Kassapoglou and Curran. A high-fidelity structural sizing method has been introduced by Wang et al. (2002) for the better prediction of structural performance of a spoiler. Kaufmann et al. (2010) studied the structural performance using a commercial FEA package. The effect of the internal structure layout on cost and weight was captured in these studies. The tools of Wang and Kaufmann can only be used to solve ad hoc problems and are not flexible enough to be reused for the structural design of similar structures due to a lack of a flexible modeling tool and a smooth link with FEA software.

A separate modelling module has been developed by Kelly et al. (2005, 2006) to generate the CAD model with realistic complexity. The modelling module is able to generate different structural topologies by changing the number of spars and ribs and their position. The weight of a composite spoiler is estimated based on the FEA sizing results, however, the model needs manual work to link the parametric model with the FEA package. This repetitive manual work inhibits optimization in the RFP phase when many structural layouts have to be analysed.

New topology optimization methods have emerged to search for an optimal topology from a large range of structural layouts. Wang et.al (2011) present a simultaneous partial topology and size optimization method for wing structures. In Wang et.al's method, the ant colony optimization method

is adopted at the topology level, and the gradient-based optimization method in NASTRAN is employed for component sizing. The geometry was pre-meshed before the optimization started. The number of internal structures is varied by deleting finite elements to represent the addition or removal of an internal structure, however, the position of the real structural members is not changed so there is no requirement to remeshed the geometry during the optimization loop.

Locatelli et al. (2011, 2012) have developed EBF3SSWingOpt which has been successfully used to perform the structural layout and sizing optimization of a wing box using curvilinear spars and ribs. The geometry is generated using PATRAN internal geometry capabilities which requires remeshing for each structural layout: however, EBF3SSWingOpt has difficulties in integrating with other discipline tools because of (1), difficulty with keeping consistency of the skin OML in the structural analysis with the one used for the aerodynamic analysis, and (2) difficulty with manipulating the aircraft geometry for extracting the geometric data for cost estimation. The tools used in EBF3SSWingOpt can capture the effects of the structural layout on weight for airframe structures with realistic complexity, however, no cost estimation method is implemented in EBF3SSWingOpt.

The aforementioned MDO frameworks mostly feature one or more of the following limitations. They lack the required modeling flexibility that to perform a through design space exploration; they make use of an oversimplified analysis model, e.g. no FEM analysis, which cannot sufficiently capture the behavior of innovative solutions; they lack the generality and flexibility required to reuse the framework for different products or to include other disciplines. The design systems built for this research presented here need to replicate the success of the MDO systems and overcome the limitations mentioned above.

2.1.4 Concurrent Engineering

The reduction of design lead time has been increasingly being considered by aircraft OEMs due to the pressure from airlines which require aircraft to be quicker to market. Concurrent engineering (CE), the practice of executing coupled development activities in parallel to reduce design time, has become the common mode of product development and has gained in importance since the late 1980s (Takeuchi and Nonaka 1986, Wheelwright and Clark 1992, Krishnan and Ulrich 2001). The term CE was first defined by the US Institute for Defense Analyses (IDA, 1986) as:

A systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is to cause the developers, from the outset, to consider all elements of the product life cycle from concept through disposal, including quality, cost, schedule, and user requirement.

In the past thirty years, since the term of CE was first proposed, extensive management research into CE has focused on the social and organizational mechanisms that enable various stakeholders to participate in the early design phases (Susman, 1992). In practice CE has been successfully applied in different industries in aerospace, notably, Airbus and Boeing, in the automotive industry, notably, Toyota and Honda, and for steel construction (Anumba et al., 2000). Many practitioners and academics believe that using a simultaneous and parallel design process can address industry's need to reduce lead time and product costs, while increasing product quality (Nevins, 1989; Liker et al. 1996; Pardessus, 2004).

It is not easy to make two coupled design tasks parallel. Terwiesch et al. (2002) found that in the tight project schedules, many engineers cannot afford to wait until all required information input is available and have to start a downstream design task "in the dark", replying *preliminary information* from the upstream design task, information that has not been fixed yet. Product design must be developed while uncertainty remains about the customer's needs, and subsystems must be specified while the interaction subsystems are still under development.

In a fully sequential process, see Figure 2.7, above, no information is released to the downstream design task until full knowledge of upstream design task is gained. The downstream design task can rely on the finalized information from upstream once the downstream design task starts. This process is symbolized in a formal release milestone in the process, this is indicated by the diamond shape in Figure 2.7.

Although overlapping two activities, Figure 2.7, bottom, reduces the total design lead time as the downstream design process can start early, this process is not without its drawbacks. The downstream design process starts from the preliminary information, instead of relying on the formal release milestone. As shown in Figure 2.7, the preliminary information available during a design process tends to be based on a low-to-medium upstream knowledge, symbolized by the lighter shaping in Figure 2.7. The earlier the downstream design process starts, the higher the risk of future design changes, especially if the outcome of the upstream design activity is hard, or impossible, to predict. In this case overlapping activities creates a need for additional engineering effort in the form of reworking design task already done (Terwiesch et al., 2002; Smith and Eppinger, 1994). Studies have shown that reworking can account for up to 50% of the engineering capability within an OEM and one third of the total development budget (Clark and Fujimoto 1991, Soderberg 1989).

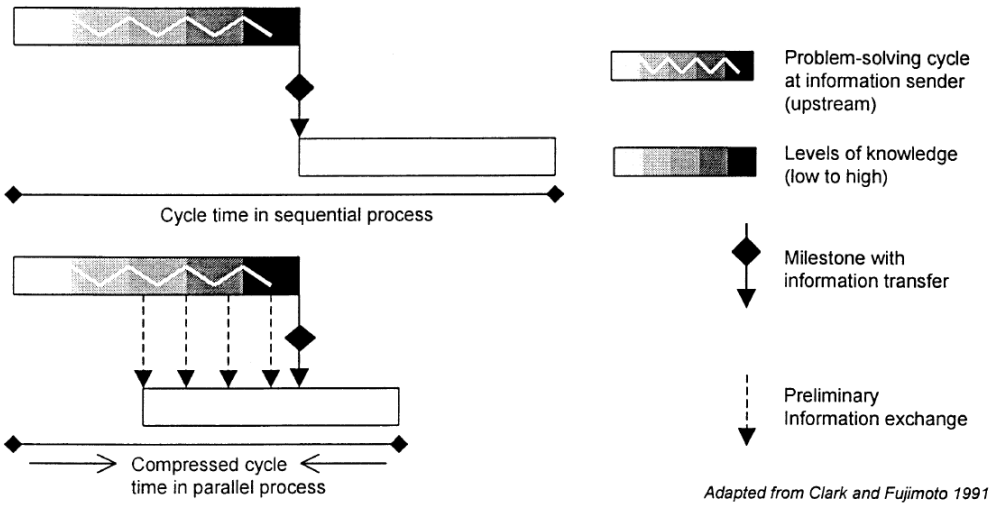


Figure 2.7: Making two sequential activities overlapped requires the use of preliminary information exchange. Adapted from Clark and Fujimoto et al. (1991)

Although airframe subsystem design is treated as concurrent with the overall aircraft design process, the overall aircraft **conceptual** design process and airframe subsystem design process are two sequential activities (Figure 1.3). In this research, involving suppliers in the aircraft conceptual design phase is equivalent to creating an overlap between these two activities. In this case the preliminary information used is very often geometry constraints and functions that the subsystem should fulfill.

Starting the local design relies heavily on the *preliminary information* that is usually only got from the global design, and requires close integration between an OEM and its suppliers.

Such close integration between an OEM and suppliers during product development is necessary to apply CE successfully (Clark and Fujimoto, 1991). This integration has been further studied by Liker et al. (1995), who state that one of the important success patterns for a supply chain is the suppliers "full service capability", i.e. the capability of a supplier solving technique problems within their subsystems without external help. "Full service capability" in the airframe design domain includes the suppliers' capability to perform local design fulfilling all the design requirements, and the capability to assume *preliminary information* when this is not available from the global design. Therefore, as this research, in part, was a search for an approach, which will allow suppliers to start early local design overlapped with the conceptual phase of the global design, understanding the capability to assume the *preliminary information* by suppliers themselves is necessary.

2.2 Airframe structural design process

The final deliverables of an airframe design will first be identified, followed by a discussion of the airframe design process. The deliverables consist of the following

- product definition, in terms of a formal design description and CAD model
- production definition, in terms of a formal description about the manufacturing concept, tools and fabrication processes
- product verification, in terms of a compliance checklist and associated compliance material: proof of strength, weight, and non-recurring and recurring costs
- supply chain

Different actors are responsible for different missions and concerns in the design process. The actors and their involvements in the design process can be described using a "swimline" activity diagram. Note: for the activity diagrams in this thesis the unified modeling language (UML) was used. Figure 2.8 is a UML activity diagram of airframe structural design, showing the interactions between OEM and suppliers in the current design process.

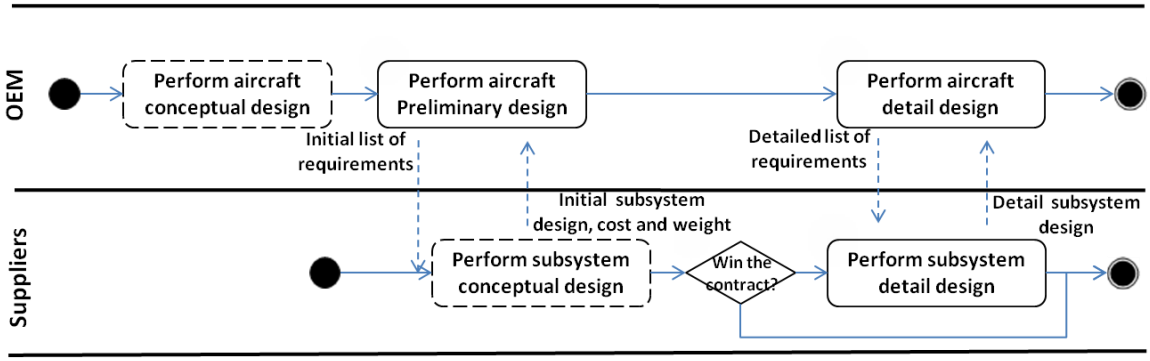


Figure 2.8: General activity diagram of airframe structural design

The OEM is responsible for providing inputs for the subsystem design, which are collected according to system requirements, i.e. airworthiness regulations and customer requirements. Suppliers are responsible for performing a subsystem design and eventually delivering the deliverables.

Additionally, after receiving the subsystem design from suppliers, the OEM is responsible for verifying whether an overall airframe design meets the system's level requirements. Since the focus of this thesis was placed on how to perform early and fast local design, the detail design phase and verification of the airframe at a global level fall outside the scope of this work. The detailed activity diagrams are given in the following sections to explain the two activities better, the dashed blocks in Figure 2.8, which are performed during an aircraft conceptual design and a subsystem conceptual design.

2.2.1 Global design process

The OEM activity diagram is shown in Figure 2.9. The required inputs need to be identified to start the airframe structural design. These inputs include the airframe OML, load sets, cost and weight budget. Load sets have to be given according to airworthiness regulations, e.g. FAR 25, whereas the OML of the airframe is usually drawn up after a three-dimensional aircraft model becomes available. The conceptual and, part of the, preliminary design of the overall aircraft have to be performed to prepare such inputs. The conceptual design of the aircraft provides an estimation of the designed aircraft's performance with the given high-level requirements of the aircraft, such as maximum take-off weight, operational empty weight, direct operating cost (DOC) and lift drag ratio. After the overall aircraft conceptual design, shown in the dashed area of Figure 2.9, is finished, the aircraft configuration has to be determined and the major dimensions of the aircraft components such as the fuselage and wing. In the preliminary design phase, the OEM aerodynamic analyst performs the 3D wing design and determines the aircraft aerodynamic properties in more refined details. The OEM load engineer selects the load sets under which the airframe should ensure the structural integrity in accordance with the airworthiness regulations. Thus, using these load sets and aircraft OML, the OEM structural engineer makes a preliminary structural design for the overall aircraft, including selecting the material to be used and the structural layout for the main aircraft subsystems, and the position and dimensions of the airframe subsystems and their interfaces. Normally, a coarse FE model of the airframe is built and analyzed for preliminary sizing for the subsystems. The load sets for designing the subsystems are often extracted from the overall aircraft FEA results, and the most promising design is selected with respect to weight and cost. After that, the requirements of more detailed performance and geometric constraints are determined for all the airframe subsystems. At this point the design process enters the overall aircraft detail design phase.

It should be stressed here that the weight of the airframe subsystems such as the fuselage and lifting surfaces have a significant influence on the weight and balance, i.e. the center of gravity of the entire aircraft, however, the center of gravity position has an influence on the calculated load sets for the airframe, hence on the airframe weight. As a result, an iterative process is required to determine airframe loads and estimate the airframe's weight.

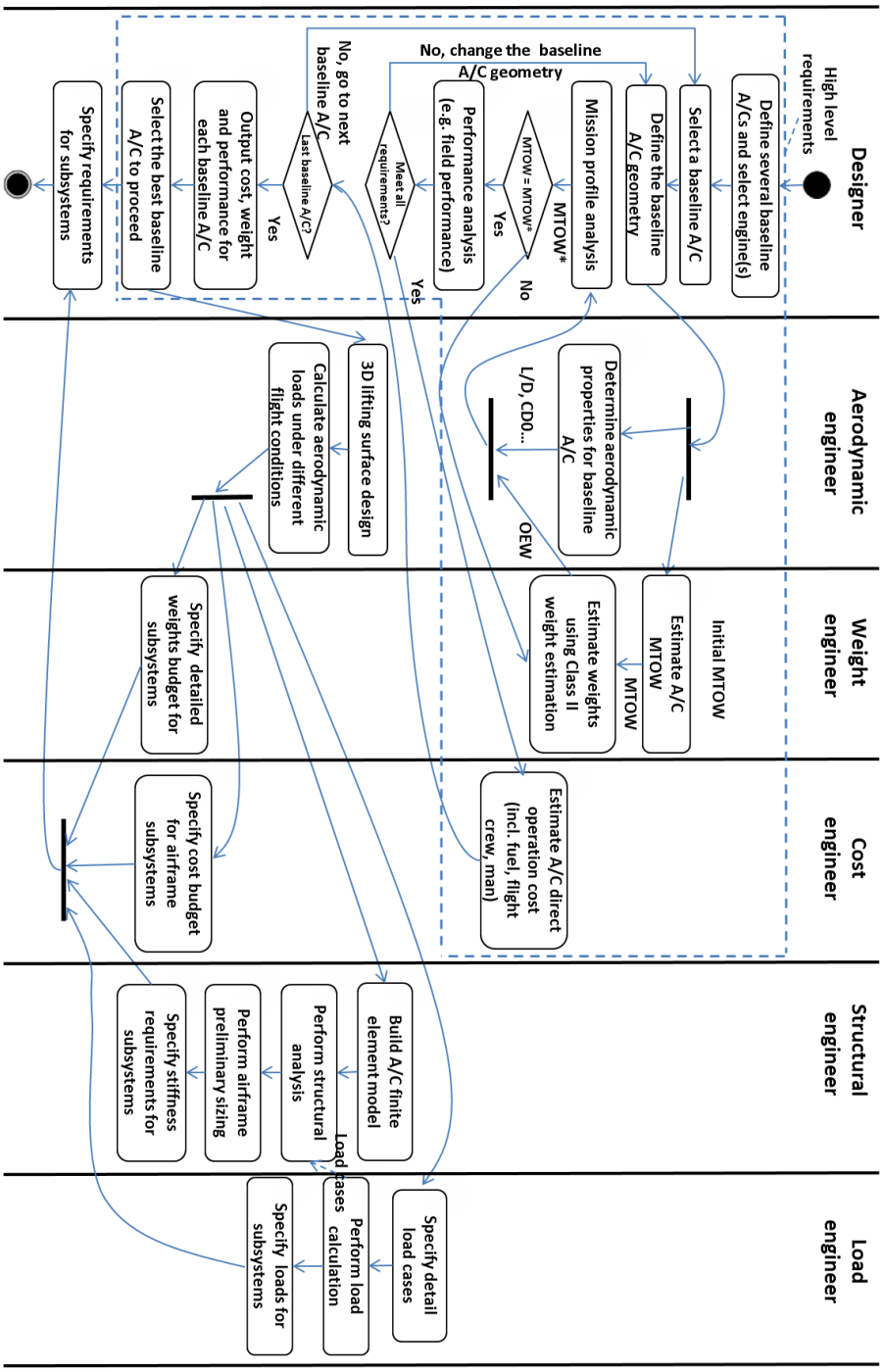


Figure 2.9: Airframe global design activity diagram, note the actors are all from OEM

2.2.2 Local design process

A supplier activity diagram is shown in Figure 2.10 to illustrate the local design process. After receiving an OML, load sets and cost and weight budgets for the airframe, the supplier designer and the supplier's manufacturing engineer will conduct a product definition and production definition respectively. A product definition will include the material used for each structural member of the airframe, the supplier of each member and its position and installation. The production definition determines how a product is built from the individual parts to the final assembly. Product and

production definitions are often documented in terms of a three-dimensional CAD model with a formal description and then delivered to other domain experts for verification.

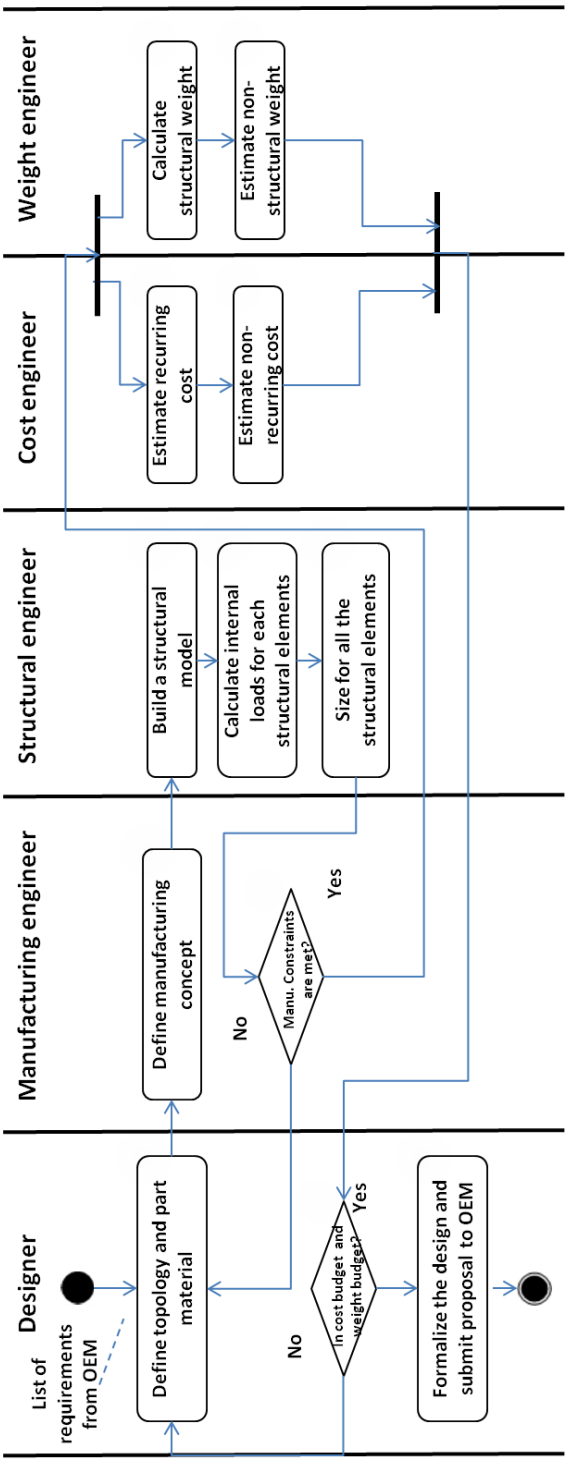


Figure 2.10: Airframe local design activity diagram (actors are all from suppliers)

To deliver a proof of strength for an airframe, the internal loads of each structural element must be calculated by the structural engineer. The margin of safety for structural elements of that airframe also has to be determined by comparing the strength with the internal loads. Very often a dedicated FE

model of the airframe is built to compute the internal loads, especially in the case of complex airframe subsystems. Cost and weight have to be calculated to verify that the design stays within budget. In the cost estimation a bottom-up method is employed once the information about the part details and manufacturing concept become available. The total weight of the airframe is also estimated using the bottom-up approach by adding the weight of all structural members, secondary structures, the fasteners and copper mesh for lightning protection for instance, are taken into account based on their material volume.

When the design of all the airframe subsystems is completed, the cost and weight of the overall airframe are calculated and verified up to the global level. It should be noted that the stiffness of the subsystems designed by suppliers will not be the same as the one assumed by the OEM. This difference leads to load redistribution. Therefore, the airframe loads and cost and weight budgets need to be then recalculated. As result, the suppliers need to modify their designs to catch up with the new loads. This process may need several iterations until the loads stop varying.

The local design process for an airframe follows the basic design cycle, the right side of Figure 2.2, as defined by van Tooren (2003). ***Specification*** is the first step, which is done by an OEM to provide a list of requirements which includes expected behaviors of an airframe such as expected cost and weight. Then, the supplier designer and the supplier manufacturing engineer carry out the second step, ***concept generation***, to generate a design concept, a formal design, in which the product and production definition are defined. The third step is ***analysis***, which is done to determine the actual behavior of the design concept and includes cost estimated by the supplier cost engineer, weight estimated by the supplier weight engineer, and internal stresses and strains calculated by the supplier structural engineer. The fourth step, ***evaluation***, is required to evaluate whether all the airframe requirements, e.g. structural requirements, are satisfied. If the requirements are not satisfied, a new design concept is generated and the above four steps repeated, until the requirements are satisfied. Finally, the supplier designer delivers a formal design to OEM.

2.2.3 Global knowledge and local knowledge

Before discussing global knowledge and local knowledge a definition of knowledge is given below (Milton, 2007).

$$\text{Knowledge is the } \begin{Bmatrix} ability \\ skill \\ expertise \end{Bmatrix} \text{ to } \begin{Bmatrix} manipulate \\ transform \\ create \end{Bmatrix} \begin{Bmatrix} data \\ information \\ ideas \end{Bmatrix} \text{ to } \begin{Bmatrix} perform\ skilfully \\ make\ decisions \\ solve\ problems \end{Bmatrix}$$

The OEM has global knowledge as shown in Figure 2.11 that is applied to provide inputs to suppliers to start local design, and the knowledge to verify whether a structural design of an overall airframe meets the requirements at a global level, requirements such as total cost and total weight of the overall aircraft. The providing of such input requires skill from the multiple disciplines involved in a global design, such as skill of aircraft conceptual design, i.e. sizing, aerodynamic analysis, load calculation, weight and balance and structural analysis of the overall aircraft design. Suppliers have local knowledge, shown in Figure 2.12, which is used to define and verify the airframe subsystem design at a local level.

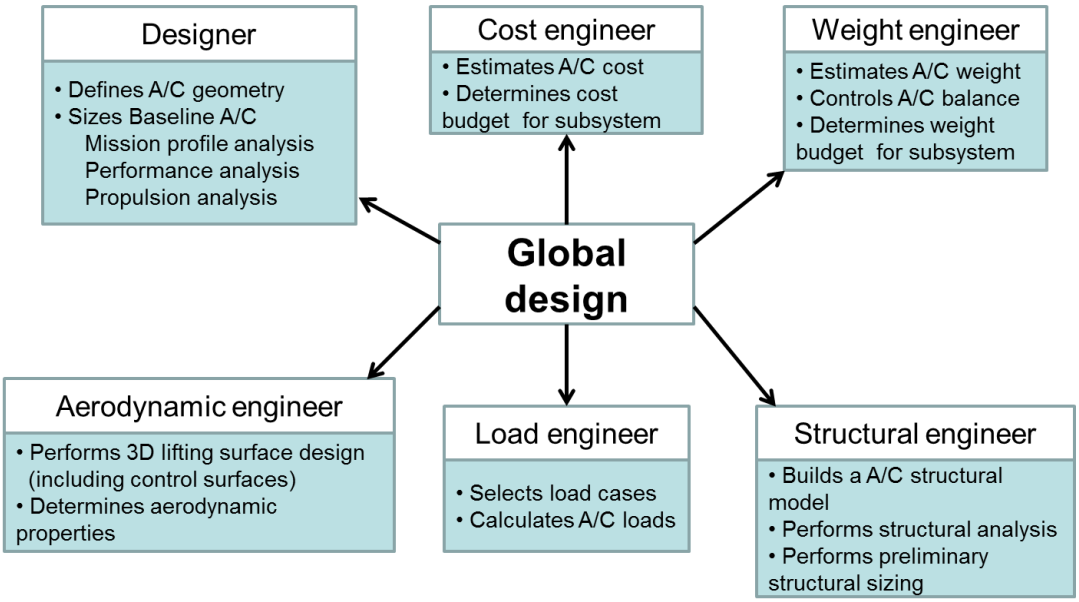


Figure 2.11: Global knowledge

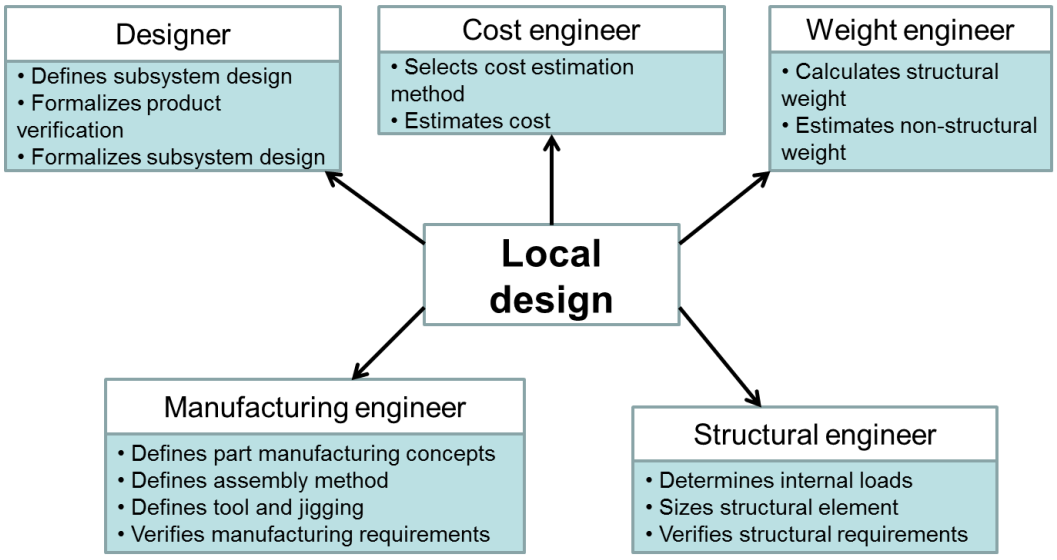


Figure 2.12: Local knowledge

2.3 Concluding remarks

In the current top-down airframe design process, an OEM works at the global level to provide inputs, e.g. cost and weight budget, load sets and geometric constraints, required for a local level design. Suppliers at the local level perform local designs according to the FBS framework, however, this top-down approach should be linked with a bottom-up approach to help the OEM make the right cost and weight estimations during the conceptual phase of the overall aircraft design. This reduces the risk of costly design changes being required in the later design phases. This top-down bottom-up approach seeks to create overlap between the local design and the overall aircraft conceptual design.

The local design and the overall aircraft **conceptual** design are two sequential design tasks, as shown in Figure 2.8. In the CE research fields, the *preliminary information* is used to enable overlap between two sequential design tasks, however, the suppliers are dependent on the OEM for provision of the *preliminary information* required to start the airframe subsystem design. Additionally, in the early global design phase, the *preliminary information* tends to change, but the local design is not flexible enough to deal with the change of the global design. For example, OEM provides suppliers with an OML, from which the product model of an airframe subsystem, usually in terms of a CAD model, and a refined FE model are built. When the global design changes, e.g. the OML changes, the CAD model and the FE model of the subsystem have to be manually re-built, which takes a lot of engineering effort.

A further issue concerns the supplier's incapacity to deliver a family of Pareto design solutions due to the slow design speed. In the current airframe design process, only a few feasible local designs can be delivered in the short design lead time. The design speed is slow mainly because of the repetitive nature of the design activities. One example of a repetitive design activity is creating different views of an airframe subsystem. The meaning of 'different views' is that the different actors involved in the design process can look at different aspects and details on the same product (Tomiyama et al., 1989). An example of multiple views of an airframe subsystem is illustrated in Figure 2.13. The repetitive model adaption and preprocessing for multiple views reduces the number of design variants that can be generated and analyzed in a given lead time, hence delivering just one feasible design.

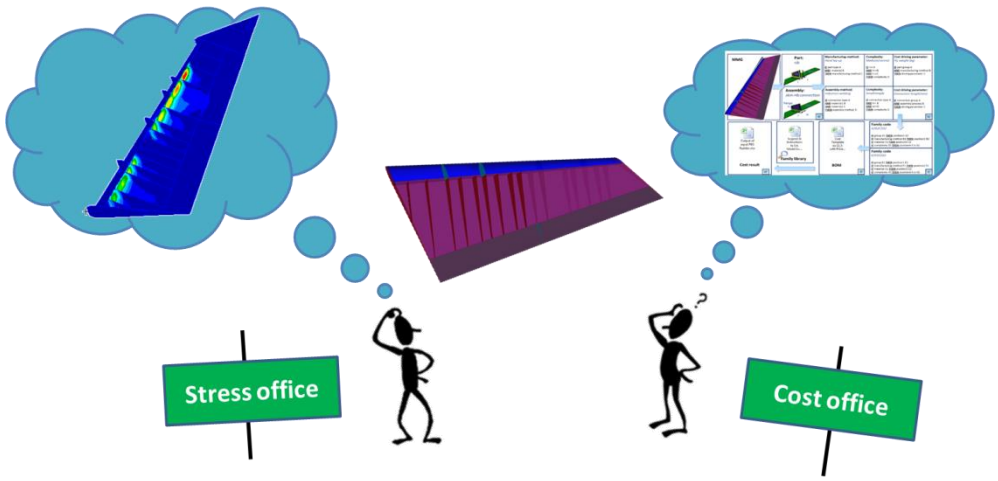


Figure 2.13: Different views of different domain experts on the same airframe component. Adapted from La Rocca (2011)

To summarize, three issues are identified in the current airframe subsystem design process:

1) suppliers' dependency on OEM to provide inputs (preliminary information) needed to start the local design
2) the local design is not flexible to deal with the changes of global design
3) the incapability of delivering a family of Pareto design solutions

These three issues can be addressed using the global-local knowledge coupling approach which is discussed in Chapter 3.

Chapter 3. Proposed approach

A global-local knowledge coupling approach that resolves the issues discussed in Chapter 2 is proposed in this chapter. Knowledge based engineering (KBE) is discussed in Section 3.2, as a support technique for multidisciplinary design optimization (MDO), which was adopted in this research for the automatic search of solution space for the design of complex airframe system. The design and engineering engine (DEE) and the KBE-inclusive MDO framework concept which embodies the proposed approach is discussed in Section 3.3. Three implementations of the DEE concept, called airframe design and engineering engines (ADEE), are illustrated in Section 3.4, namely the fuselage ADEE, the fuselage panel ADEE and the movable ADEE. An automated finite element (FE) based structural analysis method adopted for all of the three ADEEs is discussed in Section 3.5 and used to automate the repetitive pre/post-processing activities for structural analysis.

3.1 Global-local knowledge coupling approach

As discussed in Chapter 2, there are three issues that hamper fast and early local design:

1) local design is dependent on inputs provided by global design (Figure 3.1(a))
2) local design is not flexible to deal with the change of global design
3) a lack of ability to deliver a family of Pareto design solutions

The global-local knowledge coupling (Figure 3.1(b)) supports local design using capture of global design knowledge and automation of the repetitive design tasks. It comprises two modules at the global and local design levels. One is the cross-over, which captures global design knowledge and provides the input required for starting a local design. The second is the module at local design level [1] consisting of a set of parametric product and process models of airframe subsystems used to automate repetitive design actions at the local design level, such that the analysis and evaluation of subsystem designs can be quickly performed. Formal definitions of several important terms are given below for clarity.

Cross-over*: a set of tools and methods which mimics the global design process and captures global design knowledge to provide the inputs needed to start the local design.*[2]

Knowledge coupling*: suppliers who have local knowledge extend their capabilities by integrating (part of) the global knowledge such that suppliers do not heavily depend on OEM for providing inputs needed to start the local design in the conceptual phase of the global design.*

Using the global-local knowledge coupling, not all of the inputs required to start early local design are obtained from the global design. From the cross-over, suppliers can derive many of the required inputs for local design independently and proactively. With the increased independence, suppliers can start the local design earlier (Figure 3.2).

The global-local knowledge coupling approach allows the fast design space exploration since both the local design module and the cross-over can be accelerated by automating repetitive design actions

[1] This module is called "the local design module" in the reminder of this thesis.

[2] The cross-over is only a subset or a simplified version of the global design, containing a list of requirements that can be assumed and used to start an airframe subsystem design. The development of a full set of tools to simulate global conceptual design fell out of the scope of this research.

in the design process. Supplier provides the results from the fast local design, which can be a family of solutions rather than just one feasible design, to OEM at the global level such that the design confidence of the global design can be enhanced in the overall aircraft conceptual design phase.

The design framework which embodies the proposed approach has to meet three high level functionalities.

1) The cross-over must be able to quickly provide inputs required by suppliers to start local design.
2) Using the inputs generated by the cross-over, the system must allow suppliers to quickly respond to the change of global design.
3) The local subsystem design must be able to support suppliers in the exploration of large design spaces and search for optimal airframe subsystem designs.

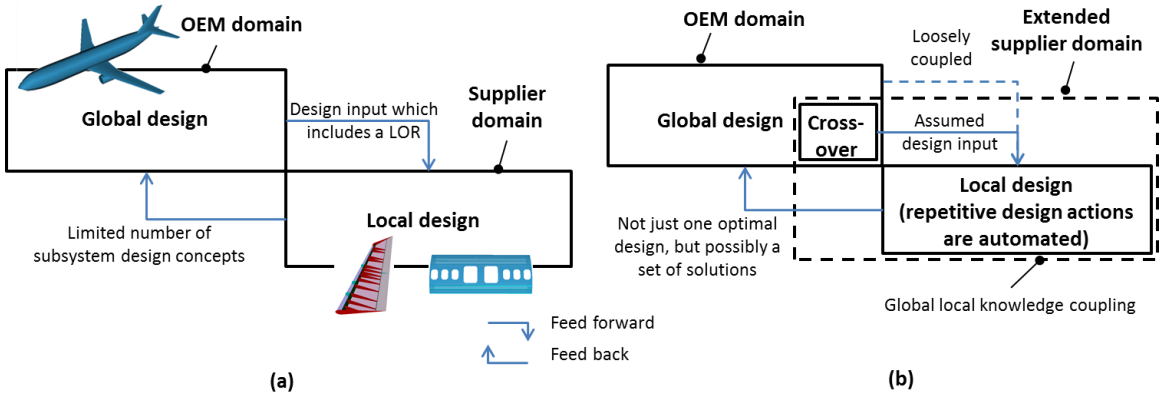


Figure 3.1: (a) Global design and local design are closely coupled in the current design process; (b) Global design and local design are loosely coupled through the cross-over

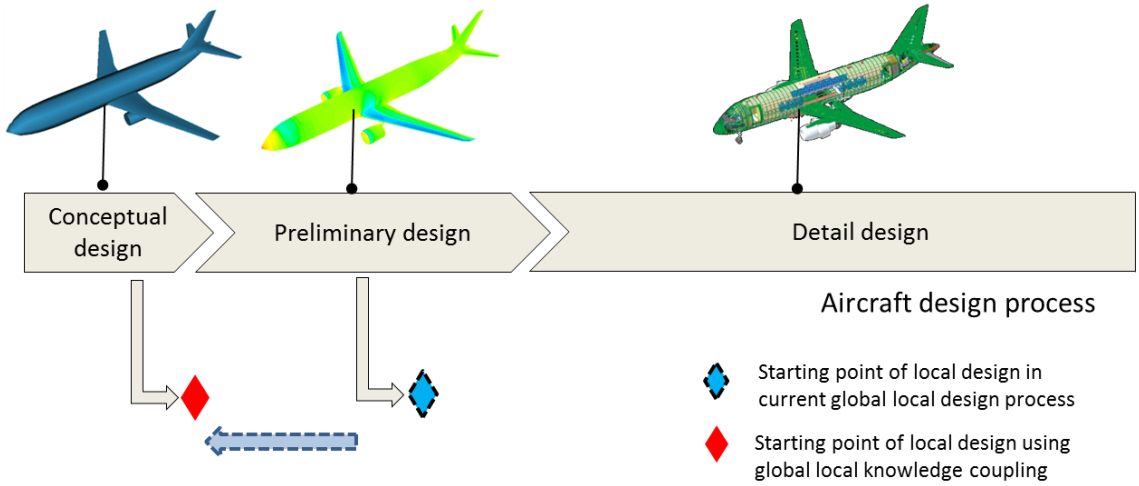


Figure 3.2: Starting point of the subsystem design is shifted forward using the proposed approach

Multidisciplinary design optimization was discussed in Section 2.1.3 as an effective mean to find the optimum design for complex systems, such as the airframe and its sub-assemblies or components. Multidisciplinary design optimization was adopted in the proposed approach to search for a set of

optimal design solutions of airframe subsystems. As stated in the research objective, the global-local knowledge coupling approach is aimed at supporting airframe local design, MDO was only adopted for local design. Performing MDO for global design was out of the scope of this research.

3.2 Knowledge based engineering

As discussed in Section 2.1.3, a generic MDO framework comprises three functional modules: namely the modeling and analysis module, the design explorer module and the optimizer (Vandenbrande et al., 2006). To enhance the flexibility and extensibility of the MDO framework, van Tooren and La Rocca (2003) split the modeling and analysis module into two separate modules: the modeling module and the analysis module.

The modeling module of a MDO framework requires at least three capabilities, flexible geometry modeling, geometry manipulation and handling both declarative and procedural knowledge. The geometry modeling capability is necessary because the analysis tools rely on the geometry, such as FEA and cost. The geometry modeling module should also be flexible enough to deal with the topology changes and product reconfiguration in the MDO process. The importance of geometry manipulation capability is amplified when high fidelity analysis is used. The modeling module must have a geometry manipulation capability to prepare the inputs for high fidelity analysis tools, e.g. aerodynamic grids for computational flow dynamics (CFD). In order to have a modeling module able to perform the preprocessing work, it is evident that the declarative and procedural knowledge must be embedded to capture and automate some of the best practices of the domain experts (La Rocca, 2010).

Knowledge based engineering is the intersection area of CAD and AI (Figure 3.3), and it appears to be the right technique to implement the modelling module of a MDO system. From the CAD field come the geometry modelling and manipulation functionalities fulfilling the needs for generation of complex products configurations and geometry pre-processing for discipline analysis. From the field of AI come the frame based expert systems (ES) which follows the object-orientated paradigm, supporting the goal of capturing and reusing large bodies of knowledge, i.e. knowledge of how to bring the right data, in the right format, to the right human designer or computer tool. Again, from the AI field comes the rule based expert system which can automate the repetitive tasks of the designer. Compared with KBE, conventional CAD systems do not have, or to a limited extent, the required capability of capturing and reusing knowledge. For an additional discussion of KBE's conceptual foundations see La Rocca (2012).

Representative successful cases of KBE can be found in the literature as well as being widely used in industry, e.g. Airbus, Boeing, EADS and Volvo. Chapman et al. (2001) have developed a KBE application to automate the modeling of automotive structures and pre-processing for structural analysis. Van der Laan (2008) adopts KBE to improve the design process of movables by: 1) automating modeling, 2) automating the model preparation for structural analysis and cost estimation, and 3) automating the data communication between disciplines. Vermeulen (2007) has developed a KBE application to support the detailed design of fiber metal laminate (FML) fuselage panels (Vermeulen, 2007). Cooper (2010) uses KBE to support the application of FML knowledge to the design of aircraft wings and the repetitive design actions are (partly) automated. Case studies are used by Cooper (2010) to show that the design framework is able to perform cost and weight tradeoffs for non-stiffened skin panels. La Rocca (2010) has developed KBE applications to support aircraft design

by automating (part of) the repetitive design actions in the aircraft modeling process. The KBE applications are used to automate the preparation of the input models for disciplinary analyses, such as input models for computer fluid dynamics (CFD) and finite element analysis (FEA). All these application cases show KBE's capabilities as the modeling module of a MDO system and its capability to eliminate waste in the design process.

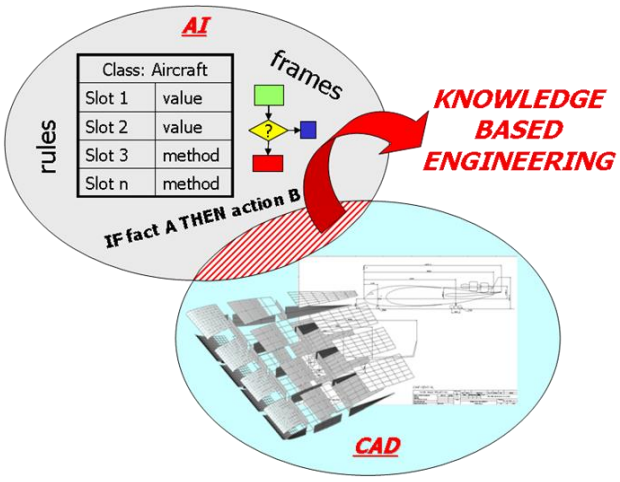


Figure 3.3: Knowledge Based Engineering (La Rocca, 2010)

As for the research work reviewed above, KBE was expected to be beneficial for the research proposed here to support the airframe design process in two main aspects. One was by providing a generative parametric airframe model that could support the generation of various structural configurations. The other was the possibility it provided to manipulate the generative model to extract data into the formats required by the analysis tools. The term "generative model" is illustrated in Figure 3.4: a set of input values is assigned to the parameters used in the product model, the KBE system applies the rules which process the input values and finally the engineered design is generated, with little or no human intervention (Cooper, 2001; La Rocca, 2011). The analysis tools used for this research included MSC.PATRAN and MSC.NASTRAN which are commercial FEA software tools (MSC, 2014) that are used for structural analysis, coupled with tools developed in-house for cost estimation.

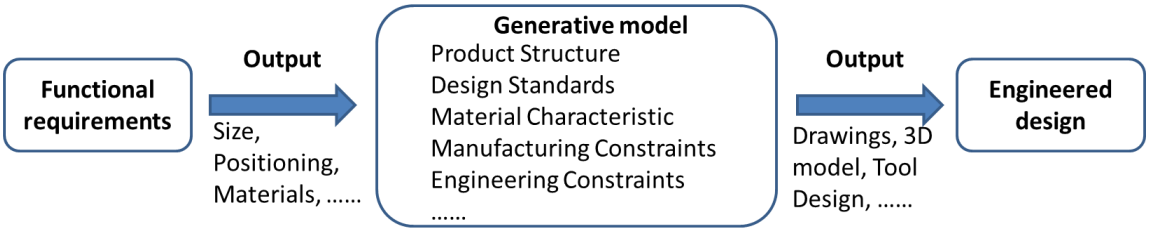


Figure 3.4: The product (or generative) model of a KBE application takes input specifications, applies relevant procedures and generates a product design automatically (La Rocca, 2010).

General-purpose declarative language (GDL) from Genworks International was the commercial KBE platform used for this research: GDL leverages the geometry modelling and manipulation capabilities offered by the SMLiB geometry kernel (SMLib, 2014). The GDL, as is any KBE system,

is able to export, and import, geometry using STEP or IGES files. In addition, GDL applications can export text files in the form of ASCII to exchange data and information between software tools.

3.3 Design and Engineering Engine

The design and engineering engine (DEE) (Figure 3.5) (van Tooren, 2003) is a KBE-inclusive MDO concept is aimed at supporting and accelerating the design process of complex products, through the automation of non-creative and repetitive design activities. The DEE is a geometry-in-loop implementation of MDO. As another view of the design cycle introduced in Section 2.1.1, i.e. specification, concept generation, analysis and evaluation, the DEE seeks to frame design activities formally to support MDO. The KBE is adopted in the DEE to generate product model and automate the repetitive activities in the MDO process. For this research, the DEE as the MDO framework concept was chosen to embody the global-local knowledge coupling approach because of:

- its capability to capture product and process knowledge for complex product design
- its quantitative nature which is enabled by its *Analysis tools*
- its flexibility to integrate new and different analysis capabilities and methodologies, which means it is able to adapt to the different nature of design cases

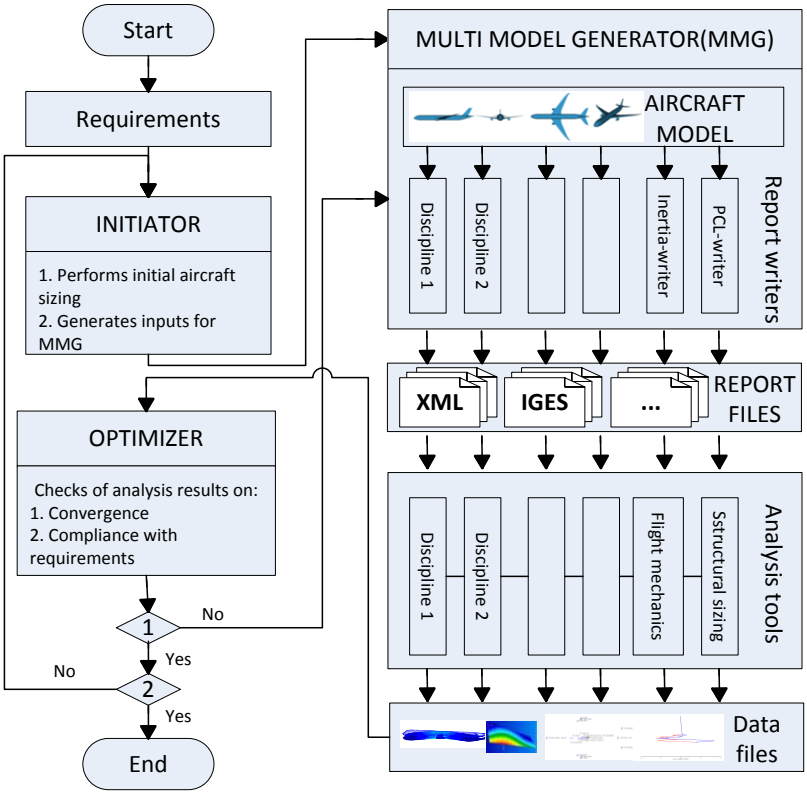


Figure 3.5: The paradigm of the DEE. Adapted from van Tooren (2003)

In a multidisciplinary design process, various designs need to be generated and analyzed to explore the large design space. Product models of these designs need to be generated and some of them have completely different configurations, e.g. different structural layouts for airframe structural design.

These designs have to be analyzed for evaluation and selection. Usually, the product model of a design cannot be directly used for discipline analysis. It is repetitive, but necessary to prepare ready-to-use inputs for discipline analysis.

The essential part of this framework is the multi-model generator (MMG) that makes the DEE different from the other MDO design systems mentioned in Section 2.1.3. In general, the MMG is a KBE application that comprises two functional parts, the high-level primitives (HLP) and the capability modules (CM). The HLPs capture product similarity and can be interpreted as parametric LEGO blocks which can be individually morphed due to their parametric definition and assembled to build up a potentially infinite range of different product configurations and variants (La Rocca, 2010). In this research, a HLP was implemented as a class generalized from a geometrical template (GT), which is a CAD primitive that provides a template for defining geometry to reduce the repetitive work for geometric definition. The class has non-geometric attributes (NGAs) that are required by analysis tools but are not represented in the geometry. An example of the NGAs of a *frame* HLP can be seen in Figure 3.6. The manufacturing method is required for the cost analysis tool and its corresponding NGA is *Manu-method*.

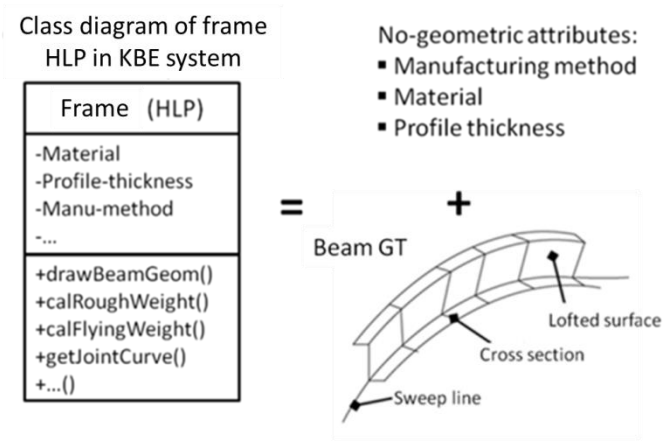


Figure 3.6: A frame HLP that contains a GT and Non-geometric attributes

The CMs manipulate the HLPs and automate (part of) the preprocessing work for various analysis tools. The CMs capture the "process similarity" and store procedural knowledge about manipulation of HLPs and preparation of inputs for analysis tools. A CM example is the *cost-inputs-collector*, which prepares an input file for calculating the frame manufacturing cost by extracting data from the instantiations of the *frame* HLP. These data are the frame height, the frame flange width, the frame profile thickness, material and manufacturing methods.

For additional discussion of the DEE's conceptual foundations see van Tooren (2003) and La Rocca (2010).

3.4 Demonstration systems

The DEE was introduced in Section 3.3 as a design framework concept to embody the global-local knowledge coupling approach. As stated in Chapter 1 the proposed approach can be applied to all airframe subsystems, however, only movables and fuselage panels of transport aircraft are used here to

illustrate this approach. Therefore, the three challenges of airframe design are discussed in the context of the design process of fuselage panels and movables of transport aircraft.

Challenge 1:

Suppliers' dependency on OEM to provide inputs, preliminary information, needed to start a local design.

In aircraft design, because of the great importance of weight estimation, a question often arises as to how the fuselage weight will change if a certain material or structural layout is adopted: to deal with this OEM often estimates the fuselage weight using semi-empirical equations in the aircraft conceptual design phase. This estimation method is called the Class II weight estimation, i.e. component weight estimation, method. An equation used in the Class II weight estimation method is given below for an Al-alloy fuselages (Torenbeek, 1982):

$$W_{fuse} = (k_1 + k_2 + k_3 + k_4) * 0.23 * {S_f}^{1.2} * \sqrt{\frac{V_d * l_t}{(b_f + h_f)}} \tag{3.1}$$

where b_f, h_f are the width and height of fuselage cabin respectively; V_d is the design dive speed; l_t is the distance between the aircraft center gravity with the aerodynamic center of the tailplane; S_f is the fuselage wetted area and k_1-k_4 are the weight penalty coefficients which can be determined as follows:

$k_1 = 1.08$ for pressurized fuselage; otherwise $k_1 = 1$.

$k_2 = 1.07$ for fuselage-attached main landing gears; otherwise $k_2 = 1$.

$k_3 = 1.04$ for fuselage-mounted engines; otherwise $k_3 = 1$.

$k_4 = 1.1$ for freight airplanes; otherwise $k_4 = 1$.

Similar empirical equations are presented by Raymer (2006). This Class II weight estimation method is not sensitive to details of material and structural layout, and is not applicable for unconventional configurations, e.g. box wing configuration.

Fuselage panel suppliers who have panel design knowledge, i.e. panel design principles, should be involved early on in the overall aircraft design process to help OEM quantify how much the fuselage weight changes if a certain material or structural layout is adopted, however, the fuselage panel suppliers are dependent on OEM to provide the inputs needed to start the panel design, which are the fuselage OML and the fuselage load sets.

Challenge 2:

The local design is not flexible enough to deal with the changes of a global design.

In the local panel design process, panel cost estimation should be sensitive to the characteristics of panel designs, e.g. skin and stringer thickness and stringer numbers, rather than relying on the empirical equations. The need to perform the design-sensitive cost estimation necessities a panel design geometry with a certain level of detail, e.g. stringers and multiple skin layers. Additionally, panel FEA is required to verify whether the structural requirements are met. The modeling of the fuselage panel and preprocessing for structural analysis and cost estimation are repetitive activities in the panel design process.

In the conceptual phase, the overall aircraft design tends to change as well as the inputs for fuselage panel design, i.e. the fuselage OML and load sets. The repetitiveness in the local design can be easily amplified due to the changes of global aircraft design, leading to the slow response to the changes.

Challenge 3:

An incapability to deliver a family of Pareto optimal solutions.

Similar to the fuselage panel design, there are repetitive activities in the design process of aircraft movables, such as movable modelling and preprocessing movable product models for structural analysis and cost estimation. Due to the repetitiveness of the design process the design speed is slow and optimization cannot, or to a limited extent, be performed in a short lead time. Suppliers can only deliver a few feasible movable design solutions.

Three instantiations of the DEE concept, called airframe design and engineering engines (ADEE), have been developed in this research work to demonstrate the global-local knowledge coupling approach. These systems are the fuselage ADEE, the fuselage panel ADEE and the movable ADEE. Challenges 1 to 3 will be addressed using the three ADEEs, shown in Figure 3.7, Figure 3.10 and Figure 3.13 respectively.

3.4.1 Fuselage ADEE

As shown in Figure 3.7, an aircraft parametric model and an aircraft FE model were developed as the cross-over to emulate a simplified version of the overall aircraft design, providing the fuselage OML and load sets needed to start the fuselage panel design. The panel sizing block shown in Figure 3.7 was used to emulate the simplified local panel design. The fuselage ADEE includes the aircraft parametric model, the aircraft FE model and the panel sizing. The ADEE is used by fuselage panel suppliers who have bottom-up knowledge, i.e. design principles and production principles for fuselage panels, to quantitatively estimate the fuselage weight for OEM in the overall aircraft conceptual design phase, considering the effects of material selection and structural layout.

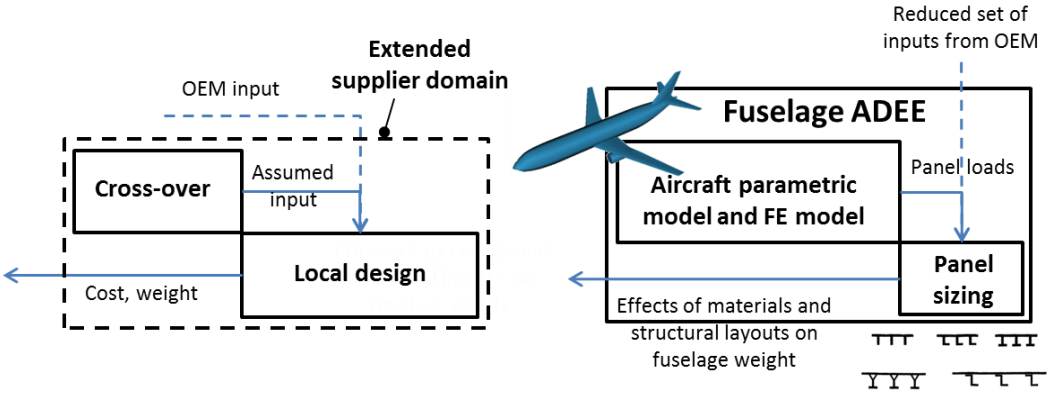


Figure 3.7: Fuselage weight study can be performed by the panel supplier, fuselage ADEE comprises of an aircraft parametric model and a global FE model as the cross-over, and the panel sizing mimicking the simplified panel local design

The total fuselage weight is decomposed into skin panel weight contribution and the weight contributions of other structures, e.g. frames, floor beams, struts, pressure bulkheads, etc.. The weight

of skin panels is calculated using a FEA-based sizing method, which uses FEA to calculate the internal loads of the skin panels and an analytical panel sizing method to determine their minimum required material. In this research, the weight contributions of all the other structures, e.g. frames, floor beams, struts, pressure bulkheads, etc., were estimated based on empirical equations suggested by Torenbeek (2013). See Appendix D for Torenbeek's empirical equations.

Using the fuselage weight estimation method described above, the fuselage ADEE must fulfill several functionalities, which are shown in the use case diagram (Figure 3.8). The fuselage ADEE will need to capture the global knowledge, such as knowledge about creating the fuselage OML, performing load calculations and structural analysis. Therefore, these functionalities have to be implemented in the fuselage ADEE.

The fuselage FEA will need to model the fuselage structures to a preliminary level of detail so that it calculates the panel loads in a realistic manner. The need for this level of detail determines which structural members are selected to be included in the MMG. The structural members included in the MMG are skin panels; frames, including the main frames; floor panels, including the cargo floors; floor beams, including the cargo floor beams and the wing fuselage intersection structures. In this ADEE, the skin panels are modeled as homogenous plates without other panel details, such as cutouts, multiple skin layers or stringers.

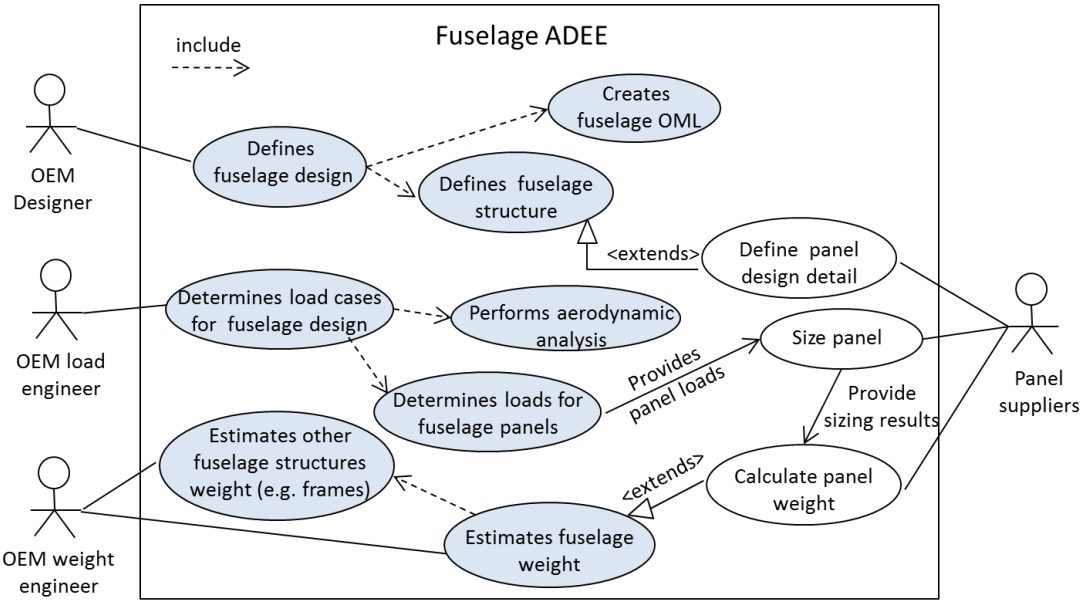


Figure 3.8: Use case of the fuselage ADEE system, global knowledge to be captured by the fuselage cross-over is highlighted in gray

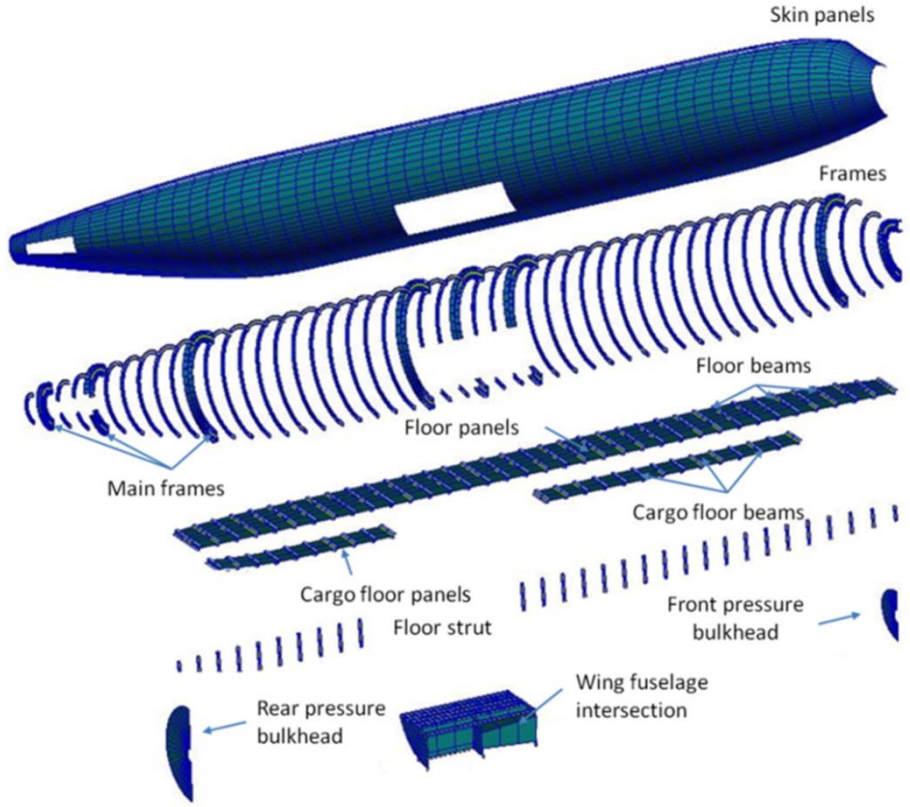


Figure 3.9: Structural members considered in the fuselage ADEE

3.4.2 Fuselage panel ADEE

As shown in Figure 3.10, the fuselage panel ADEE seeks to address the second challenge: local design is not able to quickly respond to the changes in global design.

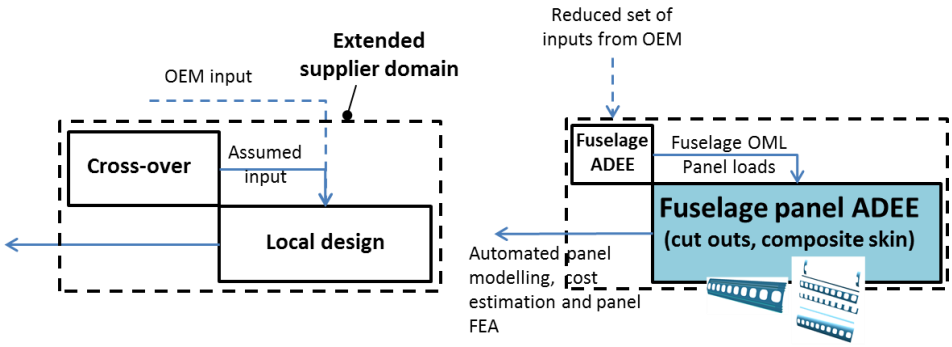


Figure 3.10: Panel design can quickly respond to the changes of the global design, the fuselage ADEE is the cross-over while the panel ADEE is developed to support the panel design

In this case, the fuselage ADEE is used as a cross-over which provides the fuselage OML and fuselage FE model containing data about load sets. The fuselage panel ADEE will address the second challenge by **automating the local design repetitive activities** to accelerate the local panel design. The activities that will be automated by the fuselage panel ADEE are panel modeling, cost estimation, weight evaluation and panel FEA.

In order to demonstrate the automation capability, the fuselage ADEE must fulfill several functionalities, shown in Figure 3.11. The fuselage ADEE will need to capture the local knowledge: this is the process knowledge about modelling fuselage panel, automating the cost estimation and automating panel FEA.

In order to feed the inputs needed for the panel cost estimation method, the fuselage panel modeling will need a preliminary level of detail so that it captures the design features necessary to feed the cost estimation model, for more details about the panel cost estimation method see Section 5.2.2. In contrast to the skin panels modeled as homogenous plates in the fuselage ADEE, the skin panels modeled by the fuselage panel ADEE comprise of frames, stringers, cut-outs and multi-layer skin such as composite skin or bonded aluminum skin, shown in Figure 3.12.

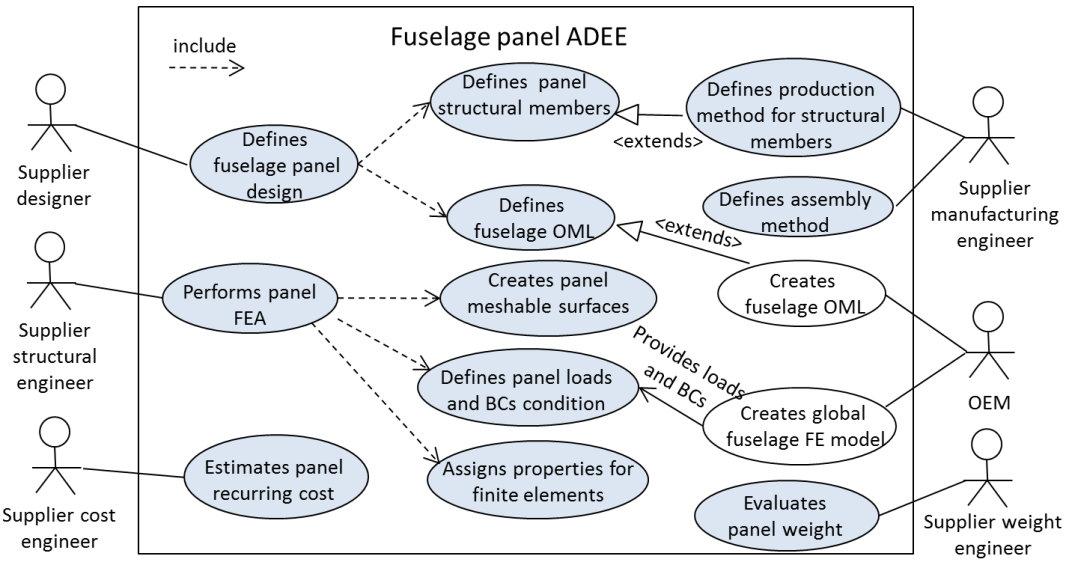


Figure 3.11: Use case diagram of the fuselage panel ADEE system, local knowledge to be captured in the fuselage panel ADEE is highlighted in gray

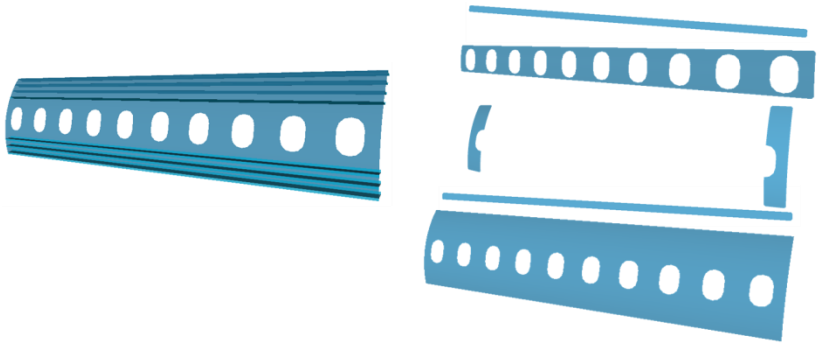


Figure 3.12: A skin panel bonded by several aluminum sheets (exploded view)

3.4.3 Movable ADEE

As shown in Figure 3.13, the movable ADEE was developed to address the third challenge: the incapability of delivering a family of Pareto optimal solutions due to the slow local design speed. The

movable ADEE seeks to deliver a family of Pareto optimal movable (local) designs rather than a feasible movable design. The verification of this ADEE was done using an optimization study of an aircraft composite movable structure.

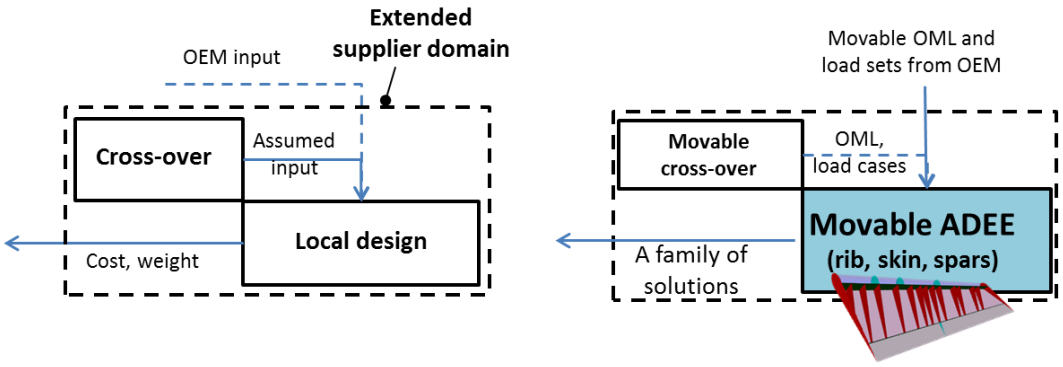


Figure 3.13: Pareto optimal solutions can be found using the movable ADEE, the moveable cross-over was developed in this research, the inputs for movable local design were directly obtained from OEM

The movable ADEE must fulfill several functionalities shown in Figure 3.14. First, the cost and weight of the product should be able to be estimated for their obvious importance. Furthermore, the cost and weight estimation methods have to be sensitive to the design variables, e.g. movable topology.

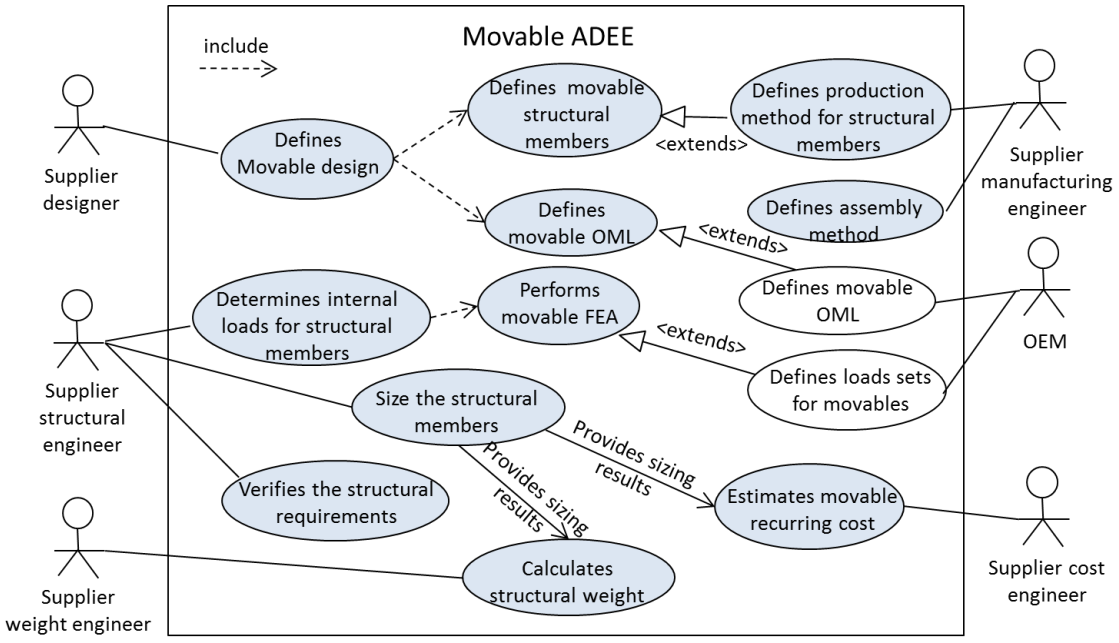


Figure 3.14: Use case diagram of the movable ADEE system, local knowledge to be captured in the fuselage panel ADEE is highlighted in gray

The movable structural members will need to be modelled to a preliminary level of detail to capture the design features needed to feed the discipline analysis, i.e. cost estimation and the FEA-based

sizing, for more details see Sections 5.2.2 and 6.2. The structural members modelled in this ADEE are ribs, spars and skin (Figure 3.15), because they bear the flight loads and have a large influence on the load path. The movable ADEE will need to support the assessment of different structural layouts, not just variants of one structural configuration, but topologically different configurations, because the structural layout will be varied in the optimization process.

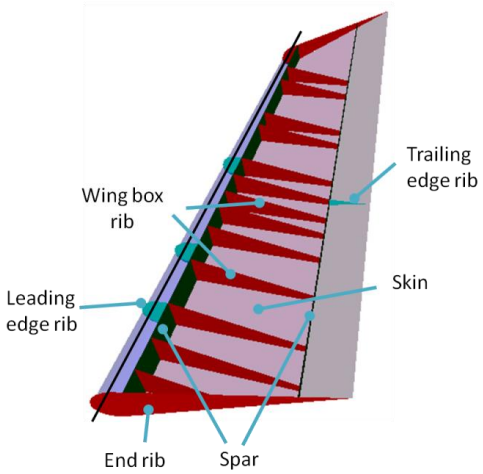


Figure 3.15: Structural members considered in the movable ADEE

3.5 Automated FE-based structural analysis

There are two main groups of structural analysis methods: analytical methods and numerical methods. The analytic methods use analytical formulas to predict the structural behaviors, i.e. stresses, strains and the stability characteristics, however, these methods are usually limited to specific applications, such as simple plates, beams and stiffened panels.

The second group of methods used for structural analysis is numerical method, usually a finite element method. In this case, a discretization of the structural domain was used, not necessarily coinciding with the actual structure of the analyzed system. The discretization often consists of a mesh, built up of points or nodes and elements used for the Finite Element (FE) analysis. The advantages of the finite element method are that the structure that is analyzed is represented by surfaces, bars, beams and solids that can closely match the actual construction (van der Laan, 2008). The analysis should therefore produce accurate results that can be mapped to the complex airframe structures.

A numerical method was adopted in this research study, however, a huge amount of work, normally repetitive, is required to set up the dedicated discretized models required by FEA software. A method was sought that could be used to link the MMG seamlessly which also generated various design configurations with FEA software for structural analysis.

Several studies have been carried out at TU Delft to automate the structural analysis of an airframe through a seamless link between the MMG and FEA software (Nawijn et al., 2006; van Hoek, 2010; La Rocca, 2011; Cooper, 2011). The methods used in the previous work and the method developed in this research are shown in Figure 3.16. Cooper et.al. (2011) and van Hoek (2010) developed a method for integral preprocessing, which means the entire preprocessing of the product model is performed

within the native KBE system before it is exported to an external FEA solver. The difference between the work of Cooper and the work of van Hoek is the meshing method. van Hoek uses the function in GDL, called “tessellation”, which generates the triangle meshes for CAD visualization. The nodes and meshes from tessellation were used for the FEA, however, the mesh quality was problematic because the tessellation resulted in many high aspect ratio triangular elements, especially at the shared edges of two surfaces (van Hoek, 2010). To increase the mesh quality, Cooper et al. developed an in-house method for quadrilateral meshing based on the work of Piegl and Tiller (1997), however, the in-housed meshing method has to be improved to handle complex airframe structures.

La Rocca (2011) and Nawijn (2006) advocate a method in which only the part of preprocessing, which is creating meshable surfaces, i.e. the surfaces ready for meshing, also called segments in the remainder of this thesis, is performed within the native KBE system while the rest is performed within FEA preprocessing software, MSC.PATRAN. Compared to the integral preprocessing, several advantages of this method are stated by La Rocca (2011):

- PATRAN's meshing capability can be fully exploited
- PATRAN's ability to generate input decks for multiple systems is directly available
- use of this method and tools can easily be applied to similar MMGs with similar outputs
- the external preprocessing ability of PATRAN that allows separate, parallel collaboration in different analysis systems of preference
- collaboration among different disciplines and companies with different software licensing structures is supported

Separating the preprocessing for structural analysis into two environments causes knowledge loss during the export of model information from the airframe MMG to PATRAN, in which only the meshable surfaces are imported and their geometric connection information can be maintained (Cooper, 2011). The knowledge of the parent structures of the meshable surfaces, material and other aspects established at the higher system level are lost. To restore the lost knowledge, the FEM-tables have been used and a stand-alone application (PYCOCO) was developed (La Rocca, 2011 and Nawijn et al., 2006). First, the surfaces within the geometric model of an airframe were chopped into surface segments, which are passed to PATRAN. The FEM-tables were exported from the MMG that included the panel knowledge reference to the segment nodes. Then, PYCOCO started reading the content of the FEM-tables. In particular, the Cartesian coordinates of each surface segment were compared with the Cartesian coordinates of the surfaces imported in the PATRAN database and, as soon as a match was found, all the information stored in the given FEM-table was automatically mapped on the corresponding representation of the surface segment in PATRAN. When the mapping was finished, PYCOCO wrote PATRAN command language (PCL) files to accomplish the pre/post processing in PATRAN.

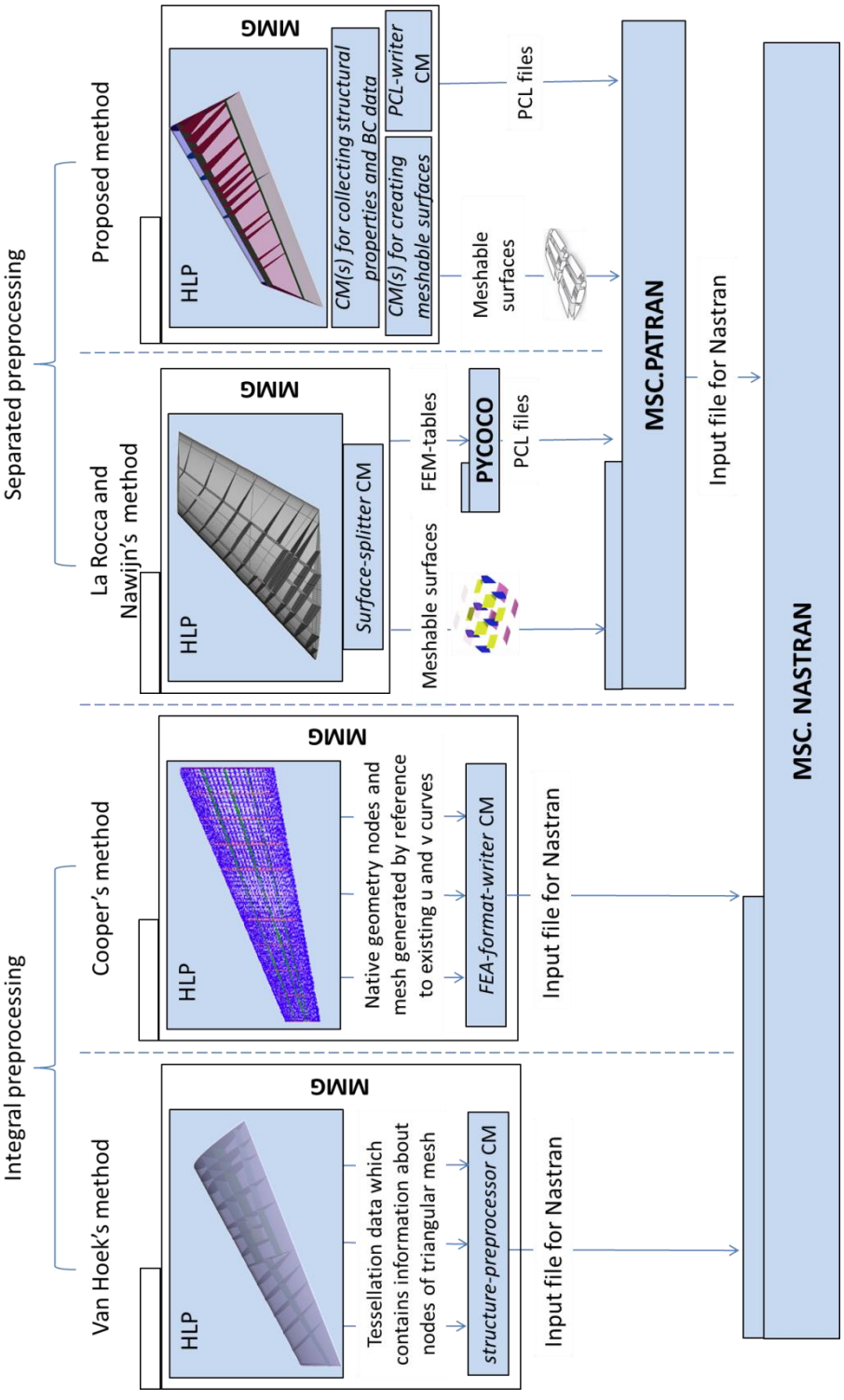


Figure 3.16: Comparison of automated structural analysis methods

Three types of data are required by the FEA pre/post processing software (e.g. PATRAN) to generate input for structural analysis: **geometry data**, **structural properties data** and **boundary conditions data**. The geometry data is needed for building the discretized FE model. The structural properties data is used to relate the material properties to the geometric entities which represent the structural members. The boundary conditions data is required to determine where an airframe structure is supported and how, and how much, the loads are applied. In La Rocca and Nawijin's method (2011), the geometry data is the meshable surfaces generated by the *Surfacesplitter* CM, whereas the structural properties data and boundary conditions data are contained in the FEM-tables, they are then exported by PYCOCO into a format that can be executed by PATRAN to accomplish the preprocessing, such as meshing and assigning properties. These three types of data are used in the rest of this thesis where the automated FE-based structural analysis is used.

La Rocca and Nawijin's method (2011) has shown the flexibility to handle different product configurations, and meantime the mesh quality is guaranteed by exploiting PATRAN's mesh capability, however, this method was developed in the currently unsupported KBE system ICAD and now is not available in GDL. This encouraged the development of a new method than the one used in the current KBE system which replicates the flexibility of handling different configurations of the older method and still generates a good quality mesh.

A new method was developed for the research (Figure 3.16) presented here. Three different sets of CMs were developed for a MMG to automate the FE-based structural analysis. The first set of CMs creates the meshable surfaces, providing geometric data for structural analysis. The second set of CMs extracts structural properties data from the product model in a similar format as the FEM-tables. The boundary conditions data are directly specified via the inputs for the product model. The third set of CMs replicates the PYCOCO's functionality of automating the preprocessing in PATRAN.

In contrast to PYCOCO, the CMs used to write the PCL files are developed within the MMG. Since the structural properties data are readily available for each meshable surface in the product model, this method does not required the mapping process used in La Rocca and Nawijin's method, which is needed to search for the properties data of each meshable surface in the FEM tables.

Creating meshable surfaces

In order to create meshable surfaces, the surfaces which represent structural members were chopped, for this research, these surfaces were chopped into smaller surfaces using a Bolean operation to keep the mesh compatibility.

Several issues of creating meshable surfaces were noticed and addressed during the implementation of the automated FE based structural analysis. In previous work (La Rocca, 2011), a requirement is posed that the segments must be triangular or quadrilateral, however, this requirement was not applied for this research because the Paver mesher in PATRAN is capable of handling the segments with more than four edges.

To guarantee mesh congruency, a constant number of grid points was assumed for each surface edge, however, high aspect ratio elements resulted from the high aspect ratio surfaces (Figure 3.17(c)). In the MMG, the concept of a "virtual element" was developed to avoid the high aspect ratio surfaces. A constant number of grid points was still used to keep the mesh congruency. An example of the virtual element is illustrated in Figure 3.18. A virtual frame was added to reduce the aspect ratio of a skin surface.

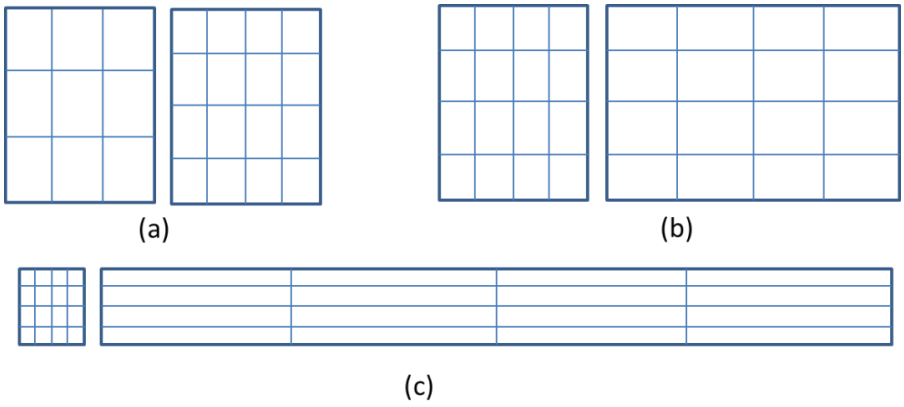


Figure 3.17: (a) Non-congruent mesh between two adjacent surfaces due to different numbers of grid points; (b)congruent mesh with satisfied mesh; (c) congruent but high aspect ratio mesh results from a high aspect ratio surface

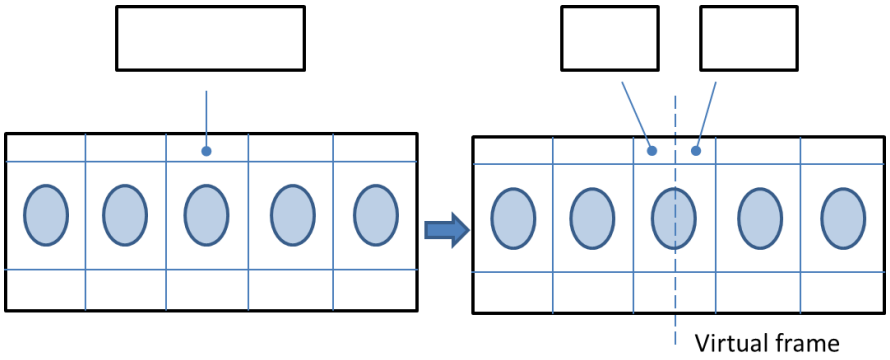


Figure 3.18: The aspect ratio of skin panel surface is decreased by adding a virtual frame

Collecting structural properties data for segments and beams

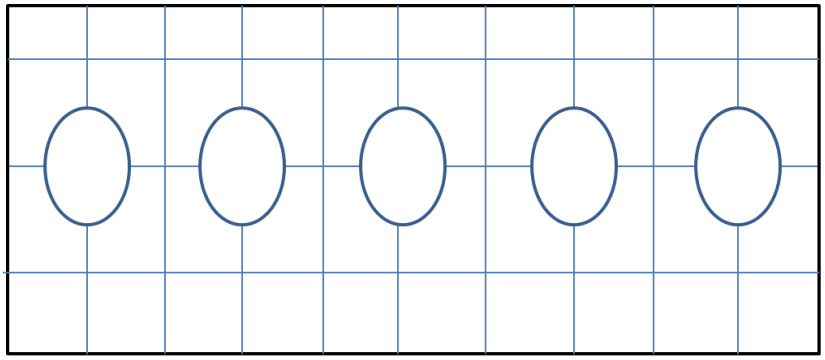
In this research, FE models of fuselage panels and movables comprised of shell elements and beam elements were used and these were discretized from segments and segment edges respectively. Therefore, two different types of data had to be derived from the product model to assign properties for these segments and segment edges in PATRAN. The two types of data were formatted in two different lists in GDL, namely the *segment-properties-data-list* and the *beams-properties-data-list*.

An example of the segment-properties-data-list is shown in Figure 3.19. Ideally, for the homogenous structures, the properties data for each segment are readily available from the segment's parent structure, the instantiations of HLPs, the material and thickness of which are specified as inputs, however, for composite structures, the segments might be covered by several plies. Therefore, mapping properties data to segments has to take into account the ply overlaps. In the fuselage panel ADEE, a mapping process is automated for this purpose, see Section 5.3.2.

The *beam-properties-data-list* contains the properties data for all the beams, because the MMGs developed in this research were surface based models, there were no curves/lines exported to PATRAN. The beam elements needed to be created in PATRAN along the segment edges. The beam properties data, section dimensions, material and section orientation, are readily available from the instantiations of the HLPs for beam like structures.

A beam-like structural member might connect with several segment edges, thus the beam properties data needs to include the beam connectivity data that indicates which segment edges should be modeled as the particular beam structural member in PATRAN. An automated process used to collect this beam connectivity data is explained in more detail in Section 5.3.2.

It should be noted here that the first and second sets of CMs differ from one ADEE to another ADEE, this means the *Surface-Splitter* CM developed to create meshable surfaces for movables is not able to create meshable surfaces for fuselage panels. More implementation details about creating meshable surfaces and collecting shell and beam structural properties data for the three ADEEs can be found in Sections 4.4.4, 5.3.2 and 6.3.2 respectively.



Laminate

```
( :composite? t
  :segment-vertices ……
  :NSM ……
  :segment-edges ……  :group-name ……
  :segment-id …… :prop-name ……
 :material-property
 (:laminate-name ……
  :ply-mat-names ……
  :ply-thicks …… :ply-orients ……
   :ref-orient …… :shell-offsets ……))
```

Homogenous

```
( :composite? nil
  :segment-vertices ……
  :NSM ……
  :segment-edges ……  :group-name ……
  :segment-id …… :prop-name ……
 :material-property
 (:mat-name ……
  :thickness ……
   :ref-orient ……
   :shell-offsets ……))
```

Figure 3.19: Properties data for meshable surfaces (NSM is non-structural mass smeared in this segment)

Automating the preprocessing in PATRAN

Several steps are required to perform the pre/post-processing in PATRAN:

1) importing geometry or modeling geometry in FE software
2) assigning properties to the geometric entities
3) meshing the geometry
4) applying the loads and boundary conditions
5) configuring the analysis problem and submitting it to the analysis solver
6) importing and post-processing the analysis results

Fulfilling these steps requires repetitive manual work via the graphical interface of PATRAN, which should be automated in the design and optimization process.

The *PCL-writer* CM developed in this research was the third set of CMs shared by the three ADEEs. The *PCL-writer* CM uses a set of string templates which generates PCL code to be executed in batch for automating the pre/post-processing activities in PATRAN.

The *PCL-writer* receives the structural properties and boundary conditions data prepared by the second set of CMs, and then generates the PCL code. An example of the PCL code is illustrated in Figure 3.20. The code is used to steer PATRAN to assign properties for surface 234. The code is generated by the gdl_create_shell(), a string template in the *PCL-writer*. The input parameters, such as shell thickness and material type, are derived from the structural properties data, *segment-properties-data-list*. A similar code is generated for all the other surfaces to assign properties. Note: the *PCL-writer* CM is described in more detail in Appendix A.

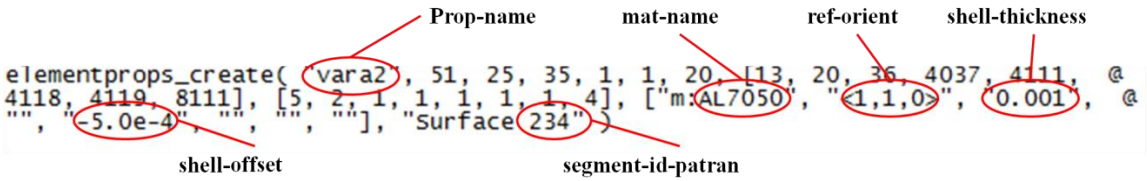


Figure 3.20: PCL code to assign shell properties for a surface.

The activity diagram of the automated FE-based structural analysis is illustrated in Figure 3.21. It can be summarized in the following steps.

1. Define a set of input parameters for the MMG, including parameters for defining the airframe geometry, parameters for defining materials and thickness for each part and parameters for loads and boundary conditions.
2. The airframe geometry is generated usingHLPs. Normally, this geometry is not ready for meshing.
3. The first set of CMs for creating surface segments chops the airframe geometry, normally surfaces, through Boolean operations to get meshable surfaces. Then, the meshable surfaces are exported using a STEP file. The second set of CMs then collects the structural properties data and boundaries conditions data from the product model, e.g. the materials and thickness, in a format required by the *PCL-writer*.
4. The *PCL-writer* CM receives the structural properties data and boundary conditions data, and writes PCL files using the PCL string templates.
5. The PCL files generated in Step 4 steer PATRAN to automatically execute the six pre/post-processing steps in PATRAN/NASTRAN.

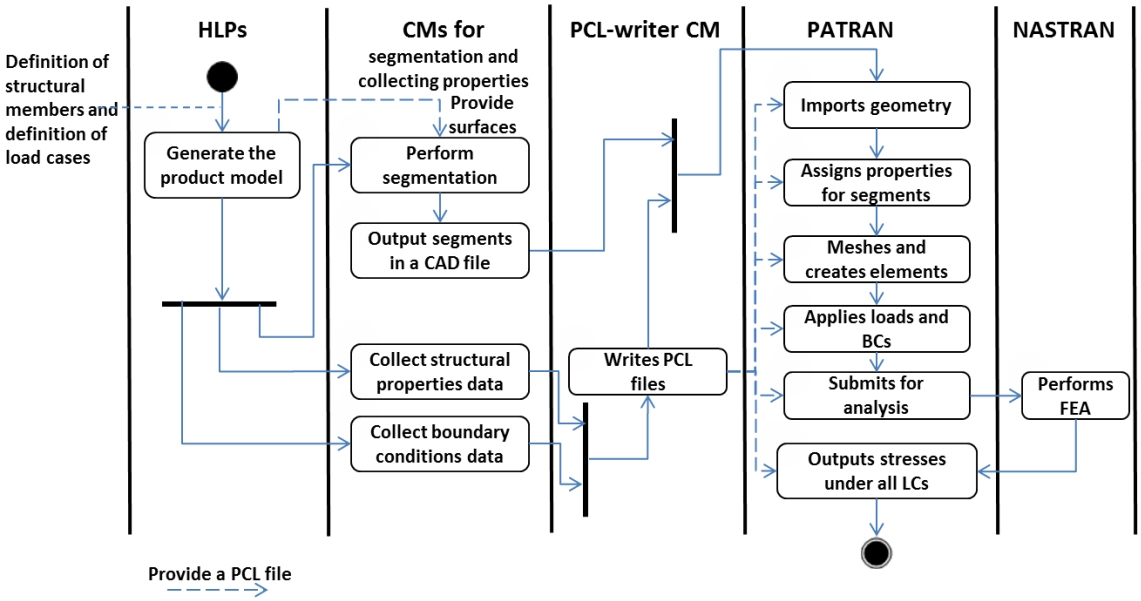


Figure 3.21: Activity diagram of the automated FE structural analysis

Limitations of the automated FE structural analysis used for this research

The automated FE-based structural analysis is not capable of handling 3D solid geometry models. In this research, the decision was made that the product model was surface based, meaning the most of the structural elements were geometrically represented as surfaces. These airframe structural members are often shell structures or plates with a thickness which is small compared to their other dimensions. The rationale behind this decision was driven by the fact that using FEA to investigate static or buckling behaviors of airframe in the early design stage does not require an explicitly modeled thickness for these structures.

Another main limitation is that the method is only applicable when PATRAN is used as the preprocessing software because the *PCL-writer* can only generate the PCL files, which cannot be used by other preprocessing software.

Chapter 4. Implementation and verification of the fuselage ADEE

The implementation details of the fuselage ADEE are described in this chapter. The fuselage ADEE was designed to be used by fuselage panel suppliers to perform design sensitive weight estimation, in which the effects of material selection and structural layout can be assessed. The outline of this chapter is as follows: an overview of the fuselage ADEE is given in Section 4.1. Then, the discipline analysis methods are discussed in Section 4.2, i.e. panel sizing method and load calculation for the sizing. The implementation details of the aircraft MMG are presented in Section 4.3. The weight estimation method was validated for a number of aircraft fuselages of different size, type and manufacturer and the results of this process are presented and discussed in Section 4.4. Some concluding remarks are given in Section 4.5.

4.1 Overview of the fuselage ADEE

The fuselage ADEE was developed to implement the fuselage weight estimation method introduced in Section 3.4.1. It consists of four main components, namely *the aircraft MMG*, *load calculation*, *FEA-based sizing*, and *fuselage weight estimation*. A schematic overview of the fuselage ADEE is shown in Figure 4.1.

Fuselage ADEE Input

The inputs required by the aircraft MMG are related to the aircraft cabin arrangement, planform parameters of the lifting surfaces, aircraft performance parameters and aircraft configuration. Additional parameters, needed for each discipline analysis tool, e.g. load calculation, in the fuselage ADEE, are defined in a text file. This includes, for example, the inputs for defining the structure materials and position.

Aircraft MMG

The *aircraft MMG* is an aircraft parametric model that has been developed within the European project AERODESIGN to support aircraft multidisciplinary design and optimization. It was used for this ADEE to generate the fuselage OML and the fuselage interior. Within the *aircraft MMG*, a parametric fuselage structural model is developed, including the fuselage skin panels and internal structures. The aircraft MMG is a KBE application, from which the inputs for each discipline analysis are derived. The reader can find more details about the *aircraft MMG* in Section 4.3.

Load calculation

In order to generate realistic load sets for the fuselage FE model, a load calculation tool was developed. The load calculation tool considers several typical load cases, such as flight maneuvering, landing impact and ground maneuvering cases. The tool was implemented in MATLAB, integrating a freeware vortex lattice code athena vortex lattice (AVL) to calculate the aerodynamic forces for the flight maneuvering cases.

FEA-based sizing

The FEA-based sizing was implemented to size the skin panels by integrating two separate modules: one was PATRAN/NASTRAN used to determine the running loads on the skin panels, and

the other was the *Panel Sizing* module, a piece of MATLAB code which was developed for this research and used for analytically determining the dimensions of each skin panel.

Fuselage weight estimation

The *fuselage weight estimation* is a piece of code implemented in the KBE system. It simply adds the weight of fuselage skin panels calculated by the *FEA-based sizing* component, and the weight of all the other structures estimated based on the empirical equations presented in Appendix D.

The implementation of the aforementioned components of the fuselage ADEE is discussed in more detail in the following sections.

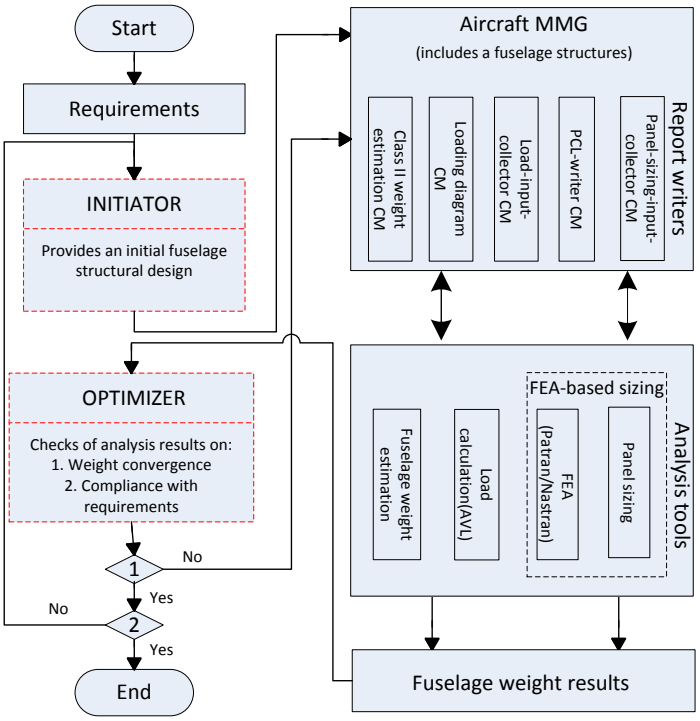


Figure 4.1: Schematic overview of the fuselage ADEE, the components in the dashed block are not implemented for this ADEE

Note: for a complete overview of the interaction and data exchange between the various components of the fuselage ADEE, the reader can refer to the N2 chart provided in Figure 4.2.

From the N2 chart, the reader can see that the weight estimation process is iterative. In this process, the *aircraft MMG* is in the centralized position. Within the *aircraft MMG*, five CMs have been developed to feed the inputs for all the three analysis tools: the *load calculation*, the *FEA-based sizing* tool and the *fuselage weight calculation*.

With regard to the *load calculation*, three CMs are required to prepare inputs for the running load calculation: these are the *Loading-diagram* CM, the *ClassII-weight-estimation* CM and the *Load-input-collector* CM. How these three CMs prepare the inputs for load calculation is explained in more detail in Section 4.3.2.

With regard to the *FEA-based sizing*, three CMs are needed to automate the FEA-based structural analysis and to size the fuselage skin panels. These are the *ClassII-weight-estimation* CM, the *PCL-writer* CM and the *Panel-sizing-input-collector* CM. The *ClassII-weight-estimation* CM is used to estimate the weight of cabin systems and equipment, which are modeled as non-structural mass in the

fuselage FE model. The *PCL-writer* CM writes PCL files to automate the FE model generation. The *Panel-sizing-input-collector* CM prepares inputs for panel sizing. How these CMs prepare inputs for the *FEA-based sizing* is explained in more detail in Section 4.3.3.

No CMs are required for preparing inputs for *fuselage weight calculation* to calculate the fuselage weight, and because this analysis tool was developed within the KBE system, it can directly extract the required inputs from the aircraft model.

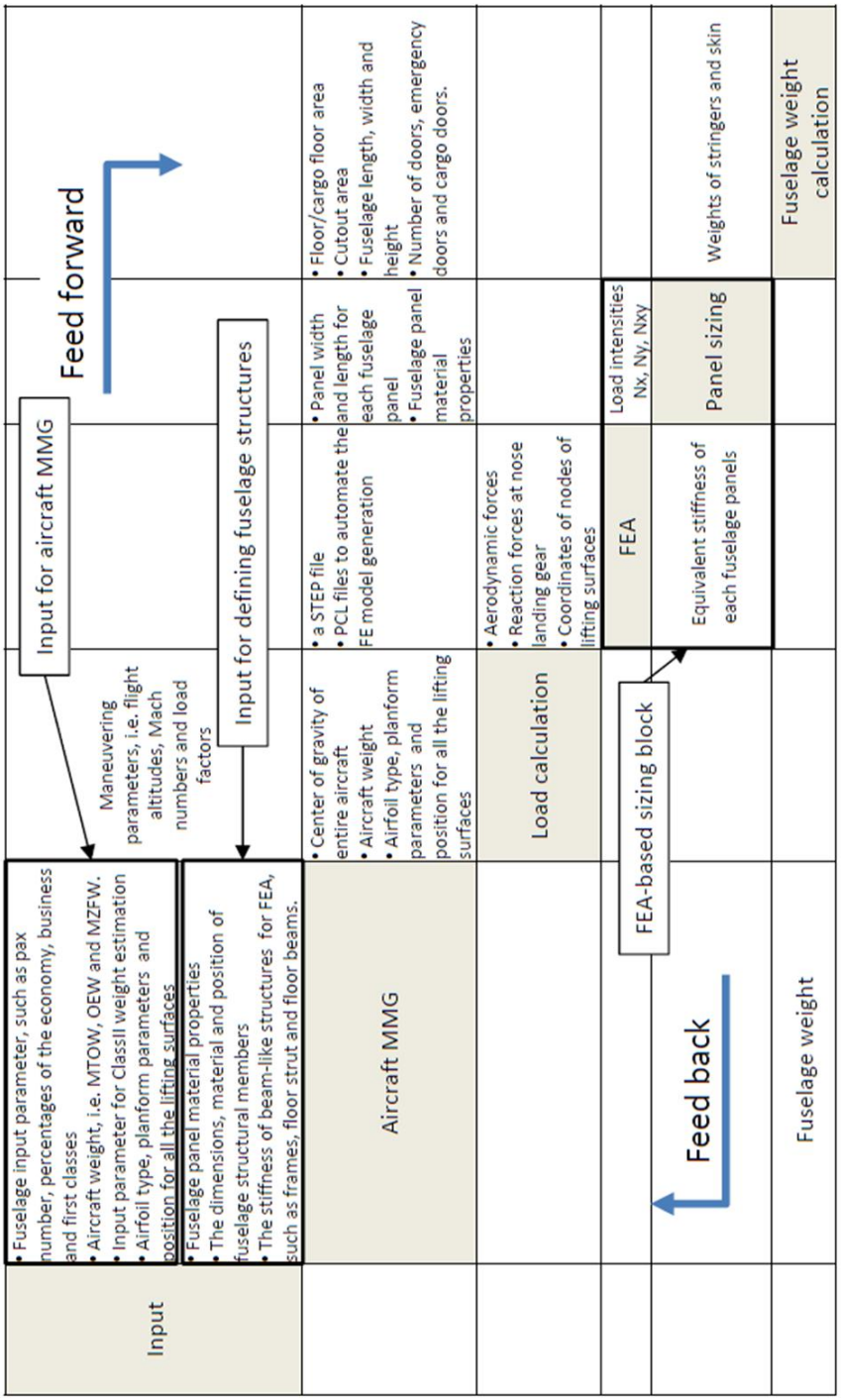


Figure 4.2: N2 chart the fuselage ADEE

4.2 Methods used for discipline analysis

4.2.1 Load calculation method

The external loads to be considered for the fuselage structural design is illustrated in Figure 4.3. Only symmetric load cases were chosen for the fuselage ADEE, including flight maneuvering cases, fatigue load cases, landing impact and ground maneuvering cases. All the load cases are listed in Table 4.1.

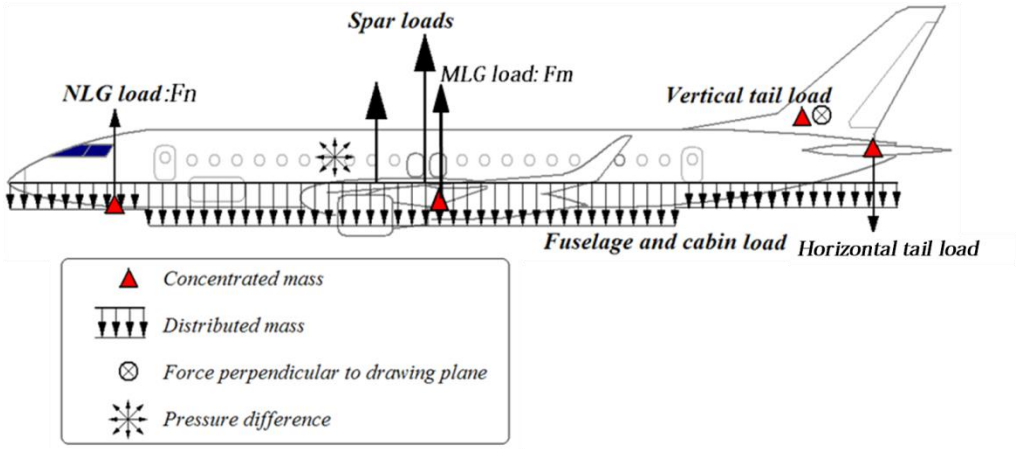


Figure 4.3: External loads for fuselage with traditional configuration

Table 4.1: Load case (LC) set used for FEA

LC ID	Load condition	Cabin pressure	Load factor	Total weight	CG position	Factor of safety
1	Flight maneuvering	1Δp	2.5 g	MTOW	front*	1.5
2	Flight maneuvering	1Δp	-1 g	MTOW	front	1.5
3	Landing	0	2.5 g	MLW	front	1.5
4	Ground maneuvering	0	1.4 g	MTOW	front	1.5
5	Fatigue	1Δp	1 g	(MTOW+MZFW)/2	(front+aft)/2	-
6	Residual strength	1Δp	-	-	-	1.15

* Front (aft) means the most forward (afterward) CG position with respect to (w.r.t.) the mean aerodynamic chord (MAC).

The normal range of the load factor for landing impact cases is from 1.75 to 2.5 for a large aircraft as suggested by Currey (1996), and 2.5 was assumed for all the validation cases presented in Section 4.4. A factor of safety of 1.5 was applied for the limit load cases, load cases 1-4.

The aerodynamic load calculation was implemented in MATLAB by integrating the freeware vortex lattice code AVL (Drela, 2008). The following inputs were required for calculating the horizontal tail balancing load:

- geometric description of all the lifting surfaces, including the airfoils, wing and movable planform
- maneuvering parameters, including flight altitudes, Mach numbers and load factors
- aircraft weight and CG position

Normally, the aerodynamic grids are not consistent with the grids used for structural analysis, and aerodynamic forces have to be mapped to structural grids. In this research, the lifting surfaces were structurally simplified as beams along the elastic axis of the lifting surfaces, as shown in Figure 4.4. Each wing was divided according to its strip location to facilitate mapping the aerodynamics forces directly on the beams. The root of each beam was applied with the lift contributed by the aerodynamic strip where the beams are enclosed. The pressure center was directly calculated from the AVL, whereas the elastic center was **assumed** at half chord of the airfoil at the specified span-wise location. Aeroelasticity was not considered for the load calculation.

For the landing load case, the reaction force at the nose landing gear was calculated by employing equilibrium equations, the summation of all the forces is zero and the summation of the moments about the CG is zero:

$$F_n + F_m = n_{lg} * W$$
$$n_{lg} = n - 1; \text{for airplane certificated to FAR part 25} \quad (4.1)$$

$$F_m * B_m - F_n * B_n = 0 \quad (4.2)$$

where W is aircraft weight, n_{lg} is landing gear reaction factor, B_m is the distance between CG and main landing gears, and B_n is the distance between CG and nose landing gear.

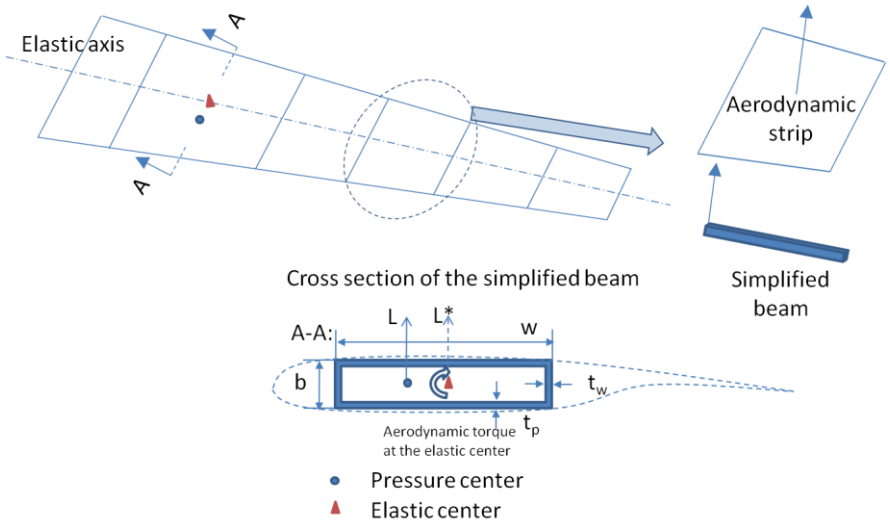


Figure 4.4: Wing structures simplified as beam elements[1]

[1] Although a wing structural analysis was not implemented in this research, the pitch moment was still extracted from the load calculation for future integration of a wing structural sizing method.

4.2.2 FEA-based sizing method

The FEA-based sizing breaks down the sizing problem into two levels: the FE model is used to calculate the running loads on fuselage panels while an analytical panel sizing method is used at the bottom level to determine the thickness of the skin panels and stringers. The FE-based sizing method can be summarized within the following steps.

1) The fuselage FE model was preprocessed, which was automated using the method discussed in Section 3.5.
2) The stress values under each load case were extracted for all the fuselage skin panels.
3) Then, using the critical stress values extracted from the FE model, the panel sizing method determined the dimensions for all the fuselage skin panels, considering the strength and stability requirements.
4) The weight of the fuselage was calculated by the *fuselage weight estimation* simply adding the weight of all the skin panels and the weight of all other fuselage structures. If the fuselage weight did not converge, the new stiffness was updated for the FE model to obtain new running loads for sizing fuselage panels in the next iteration.

The fuselage weight can be estimated by iterating steps 2 to 4 while meeting the panel structural requirements.

In the fuselage FE model, each fuselage panel is modeled as a homogenous plate with a higher extensional stiffness along the longitudinal direction of the fuselage, as shown in the right upper part of Figure 4.5. Before running the *panel size* module, the running loads in each panel under load cases 1-6 are extracted from the NASTRAN result file. These running loads are actually the stresses multiplied by the finite element thickness:

$$\begin{aligned} N_{\mathrm{x}} &= \sigma_{\mathrm{FEx}} * t_0 \\ N_{\mathrm{y}} &= \sigma_{\mathrm{FEy}} * t_0 \\ N_{\mathrm{xy}} &= \tau_{\mathrm{FExy}} * t_0 \end{aligned} \tag{4.3}$$

where σ_{FEx}, σ_{FEy} and τ_{FExy} are the stress in the longitudinal direction, the stress in the circumferential direction and the shear stress of a skin finite element. t_0 is the finite element thickness from the analyzed FE model in the previous iteration.

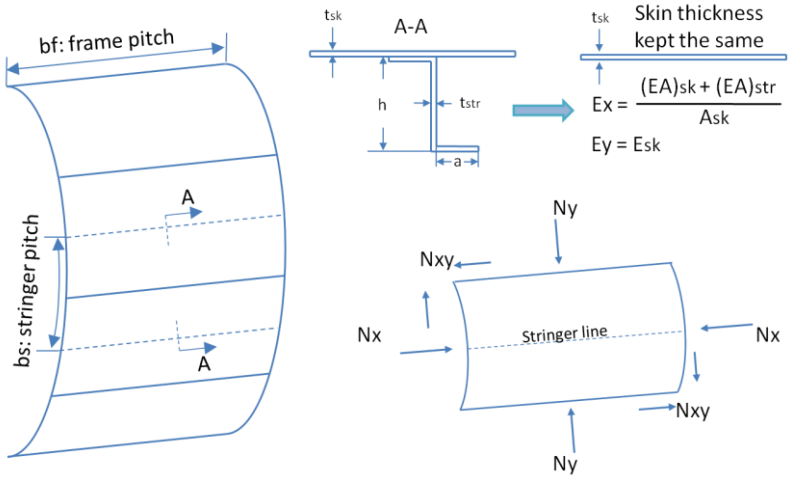


Figure 4.5: Equivalent property of fuselage skin panels

Next, step 3 of the FEA-based sizing method is elaborated to explain how a skin panel is sized. The failure modes considered by the *panel sizing* module can be summarized as follows:

- static skin material strength for load cases 1-6
  - tensile
  - compressive
- static stringer material strength for load cases 1-6
  - tensile
  - compressive
- skin buckling for load cases 1-6
  - compressive
  - shear
- stringer buckling for load cases 1-6
  - column buckling
  - stringer flange buckling
  - stringer web buckling
- stringer-skin panel Euler buckling for load cases 1-6
- skin fatigue initiation for the load case 5
- skin two bay crack for the load case 6

The stress on stringers in the fuselage longitudinal direction can be calculated according to strain equilibrium in skin and stringers

$$\frac{F_{skx}}{(EA)_{sk}} = \frac{F_{strx}}{(EA)_{str}} \qquad (4.4)$$
$$F_{skx} + F_{strx} = F_{tot} = N_x * b_s$$

where b_s is the stringer pitch shown in Figure 4.5.

The force sustained by skin F_{skx} and by stringers F_{strx} can be calculated as follows:

$$F_{skx} = \frac{\mathrm{N_x} * b_s * (EA)_{sk}}{(EA)_{sk} + (EA)_{str}},\ \sigma_{\mathrm{skx}} = \frac{F_{skx}}{A_{sk}}$$
$$F_{strx} = \frac{\mathrm{N_x} * b_s * (EA)_{str}}{(EA)_{sk} + (EA)_{str}},\ \sigma_{\mathrm{str}} = \frac{F_{strx}}{A_{str}} \quad (4.5)$$

The stress on skin in the circumferential direction can be obtained as follows:

$$\sigma_{\mathrm{sky}} = \frac{N_{\mathrm{y}}}{t_{sk}} \quad (4.6)$$

The shear stress τ_{sk} on skin can be obtained as follows:

$$\tau_{\mathrm{sk}} = \frac{N_{\mathrm{xy}}}{t_{sk}} \quad (4.7)$$

The following design criteria are used to size each skin panel.

Skin and stringer material failure

The stress values in the skin and stringer σ_{skx} and σ_{str} are checked with the allowable material compressive yield stress σ_{cuall} and tension yield stress σ_{tuall}. The fatigue initiation allowable stress $\sigma_{hoop\,max}$ is given to guarantee a stiffened panel to meet the durability requirement, and the residual strength $\sigma_{res\,cirum}$ and $\sigma_{res\,long}$ in both longitudinal and circumferential directions are checked to make sure the design complies with the 2-bay-crack criterion for damage tolerance consideration (van Dalen, 1998).

Stringer column buckling

Assuming the boundary conditions of stringers are pin ended, the stringer column buckling allowable force is given by (Kassapoglou, 2010, pp. 189):

$$F_{strall} = \frac{\pi^2 (\mathrm{EI})_{\mathrm{str}}}{{b_f}^2} \quad (4.8)$$

The stringer column buckling criteria is:

$$F_{strx} \leq F_{strall} \quad (4.9)$$

Panel Euler buckling

The overall stability of the skin-stringer panel between two frames is derived from the Euler buckling load of a stringer-skin panel with a cross section of one stringer and a skin bay:

$$F_{Euler} = \frac{\pi^2 (\mathrm{EI})_{\mathrm{panel}}}{{b_f}^2} \quad (4.10)$$

The panel Euler buckling criteria is:

$$F_{tot} \leq F_{Euler} \quad (4.11)$$

Skin bay interaction buckling

The skin buckling stress σ_{skx} has to be checked with the skin bay compression buckling allowable (Niu, 1997):

$$\sigma_{\text{skball}} = k_\sigma \frac{\pi^2 E_{sk}}{12(1-\nu^2)} \left(\frac{t_{sk}}{b_{skbay}}\right)^2 \qquad (4.12)$$
$$b_{\text{skbay}} = b_s - a$$

where k_σ is compressive buckling constant and ν is Poisson's ratio of skin material, b_s is the stringer pitch and a is the stringer flange.

The shear buckling allowable stress is given by (Niu, 1997):

$$\tau_{\text{skball}} = k_\tau \frac{\pi^2 E_{sk}}{12(1-\nu^2)} \left(\frac{t_{sk}}{b_{skbay}}\right)^2 \qquad (4.13)$$

where k_τ is the shear buckling constant.

Assuming no skin bay elastic shear buckling under 40% of the ultimate loads:

$$0.4\tau_{\text{sk}} < \tau_{\text{skall}} \qquad (4.14)$$

The skin bay interaction buckling criteria is assumed to be:

$$\frac{\sigma_{\text{skx}}}{\sigma_{\text{skball}}} + \frac{\tau_{\text{sk}}}{\tau_{\text{skball}}} < 1 \qquad (4.15)$$

Stringer web and flange buckling

Similar to the skin bay, the stringer web and flange are checked to prevent the plate compression buckling using Eq. 4.12.

Remark: These isotropic formulas can be used for the dimensioning of the metallic stringer-skin panels. See Elham et al. (2013) for the formulas for composite stringer-skin panels.

Dimensioning procedure

A program for dimensioning the skin panel is carried out by the software MATLAB. In the program, the stringer and skin thickness, stringer pitch, stringer web height and stringer flange width are varied to obtain the minimum weight per unit area. The sizing procedures of the skin-stringer panel are shown in Figure 4.6.

This panel optimization method was validated with four cases: two different optimization start points under two different design loads. The test cases indicated that the optimization method was stable for the presented panel sizing method. The four validation cases are presented in Appendix C.

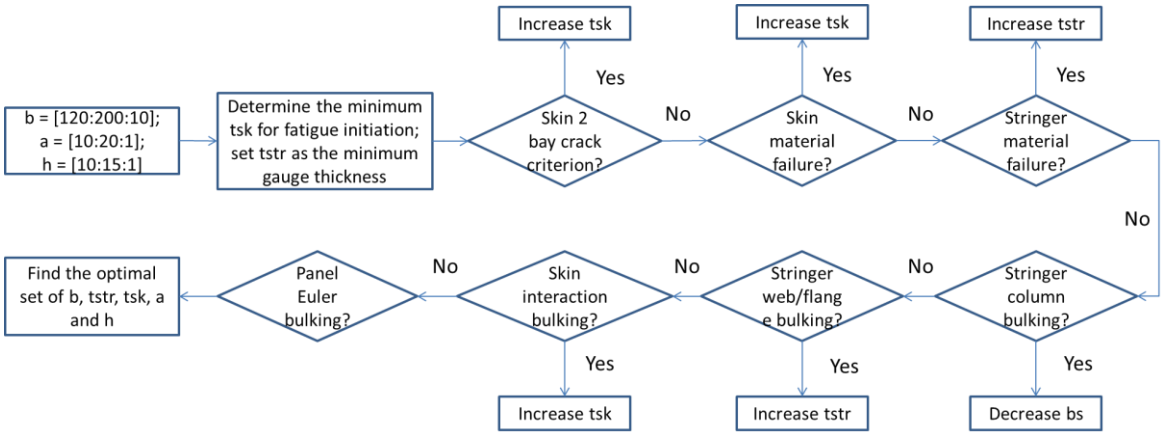


Figure 4.6: Panel sizing procedure

4.3 Implementation of the fuselage-structures class

An aircraft MMG, which is an aircraft parametric model, was developed in the European project AERODESIGN based on the methods described by La Rocca (2010). A *fuselage-structures* class was newly developed within the aircraft MMG to model the fuselage structural members and it extracts the inputs for the analysis tools of the fuselage ADEE. The class diagram of the aircraft MMG is shown in Figure 4.7 and the newly developed modules are highlighted in dashed blocks.

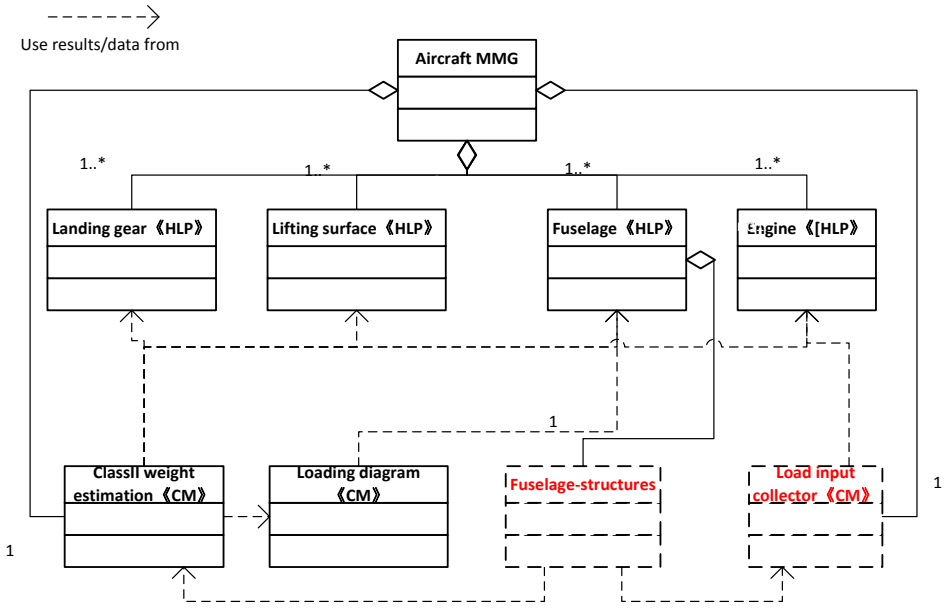


Figure 4.7: UML class diagram of the aircraft MMG

The dependencies between the HLPs and CMs of the aircraft MMG are described below.

- The *fuselage* HLP generates the fuselage OML and cabin interior, including seat arrangements, cargo containers and lavatories. The *fuselage* HLP provides inputs required by the *Class II-weight-estimation* CM to estimate the tailplane weight, equipment weight, system weight and weight for operational items. For more implementation details of the *fuselage* HLP see La Rocca (2010) and Brouwers (2011).

The positions of the fuselage items, seats and lavatories for instance, are provided to the *Loading-diagram* CM to compute the range of the CG position.

- The *Lifting-surface* HLP generates the lifting surface OMLs from which the wing area, span and sweep angle are extracted to the *Class II-weight-estimation* CM for estimating the wing/tail weight.
  The positions of the lifting surfaces are provided to the *Loading-diagram* CM.
- The *Landing-gear* HLP generates the geometry of the landing gear.
- The *engine* HLP generates the geometry of the engine and provides the CG position of engines to the *Loading-diagram* CM.
- The *Class II-weight-estimation* CM uses the Torenbeek Class II weight estimation method (Torenbeek, 1982) to estimate the component weights, such as the weight of the horizontal and vertical tail and landing gear. These weights are required to model the non-structural mass in the fuselage FE model.
- The *Loading-diagram* CM generates a loading diagram, including the CG of the entire aircraft, used for load calculation. The information about the fuselage interior is required to calculate the CG. An example of the loading diagram, in which the most forward and afterward CG positions (xcg_front and xcg_aft) can be found for load calculation, is shown in Figure 4.8.
- The *Load-input-collector* CM collects the load calculation results from the *load calculation* component of the fuselage ADEE, such as the tail balancing loads under flight manoeuvring cases and nose landing gear reaction forces under grounding or landing cases.

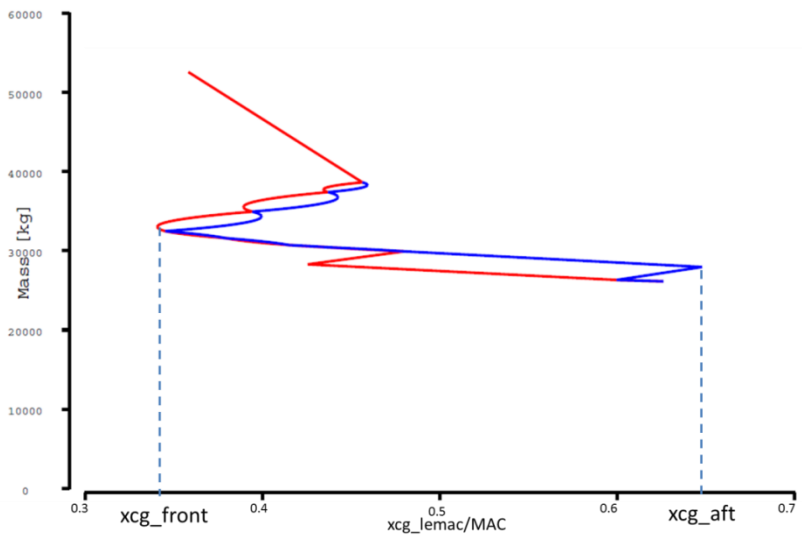


Figure 4.8: An example loading diagram generated by the *loading-diagram* CM

Requirements of the implementation of the *fuselage-structures* were identified below.

- The *fuselage-structures* should be able to model the fuselage structural members that have a large influence on the running loads. The structural members are: skin panels, frames, including the main frames; floor panels, including the cargo floors; floor beams, including the cargo floor beams, and the wing fuselage intersection structures.

- The *fuselage-structures* must be able to model the fuselage of different structural configuration, e.g. different frame pitch and number of struts, because the fuselage ADEE has to assess the effect of the structural layout of the fuselage on its weight.
- The *fuselage-structures* must be able to provide ready-to-use input files to automate the FEA-based sizing. The input files required for FEA have to be prepared by the *fuselage-structures* in a way that the fuselage FE model can be automatically generated by PATRAN/NASTRAN.

4.3.1 HLPs of the fuselage structural members

The *fuselage-structures* class is developed within the *aircraft MMG* to allow the definition of the internal structural members. The class diagram of the fuselage MMG is shown in Figure 4.9. The HLPs of fuselage structural members are defined in the following content.

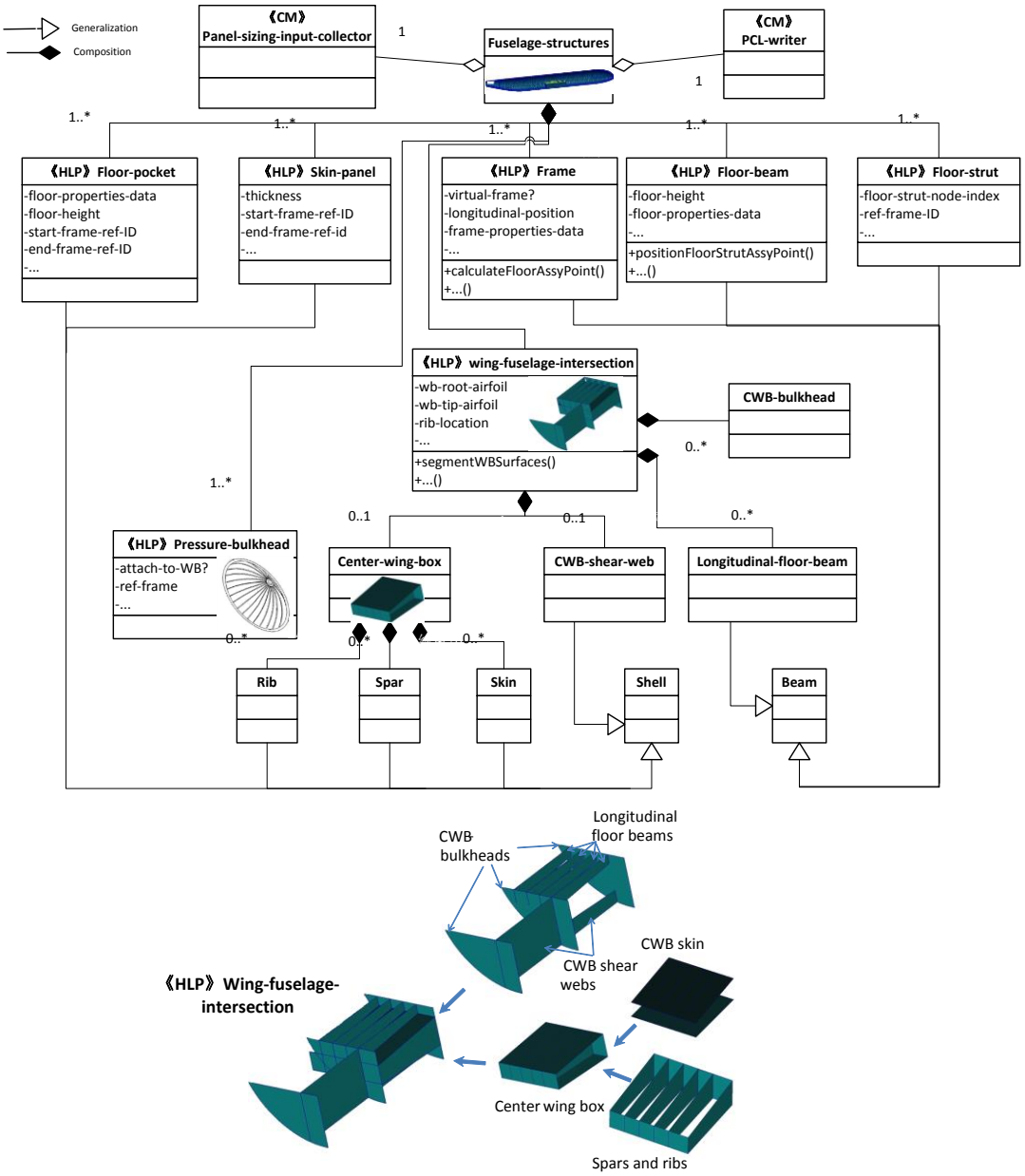


Figure 4.9: UML Class diagram of the fuselage MMG

<u><*HLP*> *Frame* and <*HLP*> *Floor-beam*</u>

A frame is geometrically represented as a curve, called the frame line, shown in Figure 4.10. It is generated by intersecting the fuselage OML with a frame cut plane, which is defined by specifying a point on the cut plane and the plane normal vector, because only symmetric load cases are considered, the fuselage FE model is a half one. As a result, only half of the frame line is needed to represent a frame. Structural properties data such as the frame axial stiffness and bending stiffness are added as non -geometrical attributes (NGAs) of the *frame* HLP. The cross section parameters and frame material properties have to be specified as NGAs when the stiffness values are not specified.

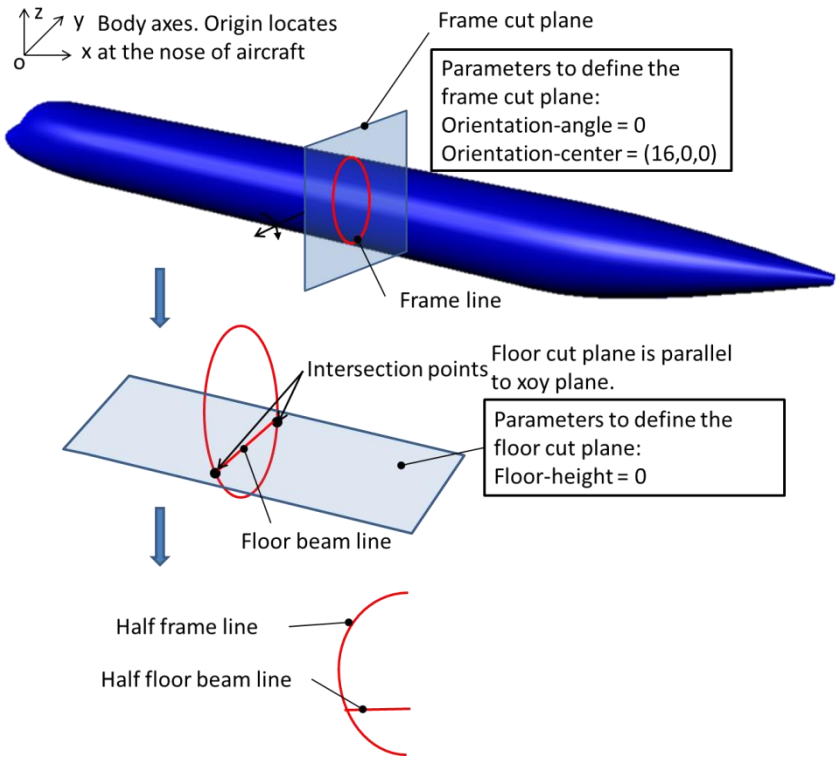


Figure 4.10: Generation process of geometric entities representing a frame and floor beam

Both real frames and virtual frames can be defined in the fuselage MMG. The real frames include normal frames and main frames. The main frames are frames located in areas of the fuselage that receive large external local loads from other aircraft components such as landing gear and lifting surfaces. These main frames have larger dimensions than the normal frames. The frames can be specified as virtual frames for the following purposes:

- positioning the non-structural items for the fuselage, such as the tail plane modeled as a concentrated mass attached to the frames
- positioning the center wing box while making sure that the mesh in the center wing box is congruent with the mesh in the fuselage.
- providing reference to define the floor and cargo floor
- defining the regions with different material properties.
- refining the mesh sizes

A floor beam is geometrically represented by a line called the floor beam line, whose start and end points always attach to a frame. The start and end points are obtained by intersecting the frame line (see Figure 4.10) with a floor cut plane whose position is user defined. Similarly to a frame line, only half of the floor beam line is needed for building a half fuselage FE model. The definition of structural properties data for floor beams is done in the same way as that for the frames. The generation process of geometric entities representing a frame and floor beam is illustrated in Figure 4.10.

<HLP> Skin-panel

The skin panels are defined after the definition of the frames. There are two sequential steps for defining a skin panel.

- One, specify two adjacent frames between which the skin panel is defined. The frame lines of these two frames are intersected by the two floor beam lines. Four sub-curves are obtained: the upper curves are above the floor beams and the lower curves below the floor beams (seen in Figure 4.11).
- Two, generate control points on the sub-curves to define the skin panel. These points are evenly distributed on the sub-curves to guarantee the surface congruency with other skin panels. These points are used as vertices of the skin panel. Another parameter has to be provided to determine the circumferential position between the two frames. The generation process of skin panels is depicted in Figure 4.11.

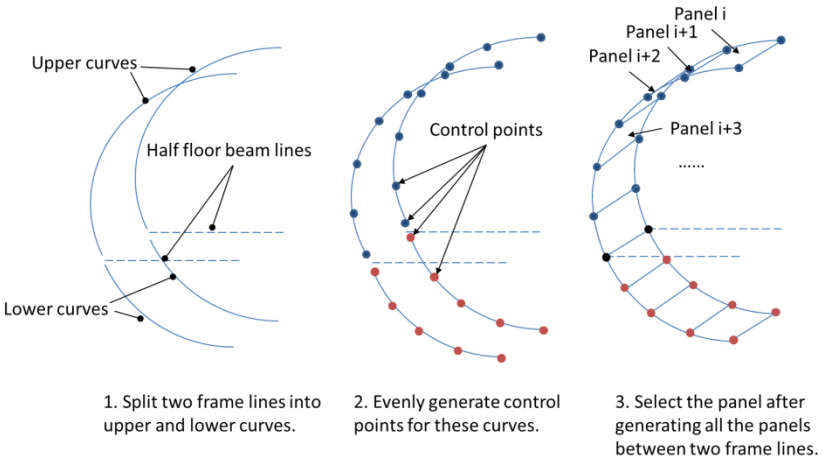


Figure 4.11: The generation process of skin panels

The skin panels are referenced in the KBE system by specifying two parameters: one parameter that indicates the skin panel's longitudinal position, and another parameter which indicates its circumferential position. This reference system for skin panels makes it possible to directly assign/update properties during the structural analysis. The limitation of this definition method is that only quadrilateral skin panels can be defined, hence triangle panels, for example, cannot be modeled using this HLP.

<HLP> Floor-panel

The definition method used for of a floor pocket is similar to that used for a skin panel. One, specify two curves representing two adjacent floor beams between which the floor pocket will be defined. Two, generate control points in a way that they are evenly distributed on the two curves to guarantee the surface congruency with other floor panels. A collection of quadrilateral panels results and one of these panels can be selected by specifying a parameter that indicates the position of the floor panel within the collection. An example input parameters for defining a floor panel is given in Figure 4.12.

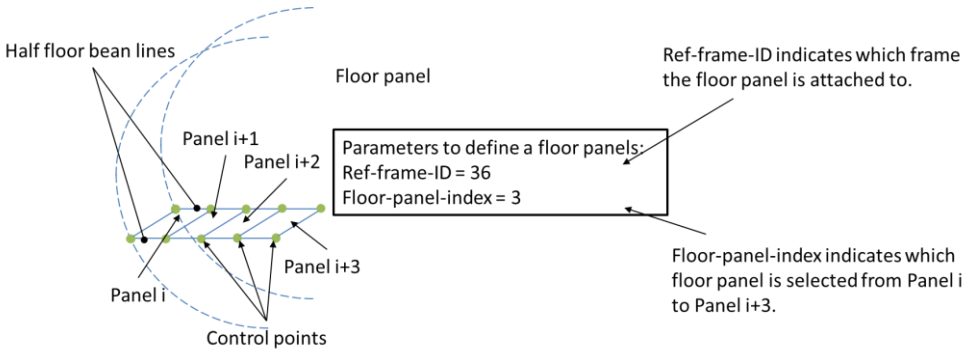


Figure 4.12: Example input parameters for defining a quadrilateral surface representing a floor panel

<HLP> Floor-strut

A floor strut line is used to represent a floor strut geometrically. The start point of a floor strut line is always selected from the control points on a half frame line whereas the end point is obtained from the control points on a half floor beam line. An example of input parameters for defining a floor strut line is shown in Figure 4.13.

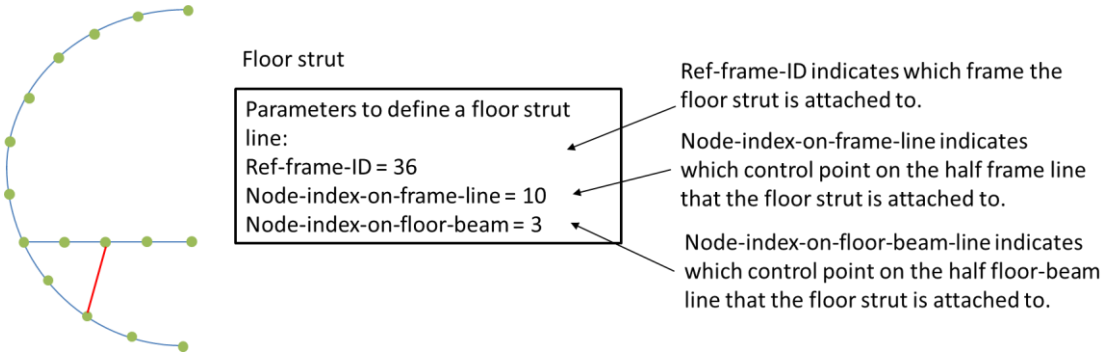


Figure 4.13: Example input parameters for defining a floor strut line

<HLP> Wing-fuselage- intersection

The modeling process for the fuselage and wing intersection was based on the method described by La Rocca (2010). Corresponding to the front and rear spars of the wing box, two cut planes are created to intersect the fuselage OML and the resulting surfaces are used to model the wing-attached bulkheads. The third pressure bulkhead is created in a similar way to close the landing gear bay. The center wing box was created based on the method that is used to model wing trunks by La Rocca (2010). An example geometry of a generated wing fuselage intersection for low wing configuration aircraft is shown in Figure 4.14. For details of the generation method of the wing fuselage intersection structures see La Rocca (2010).

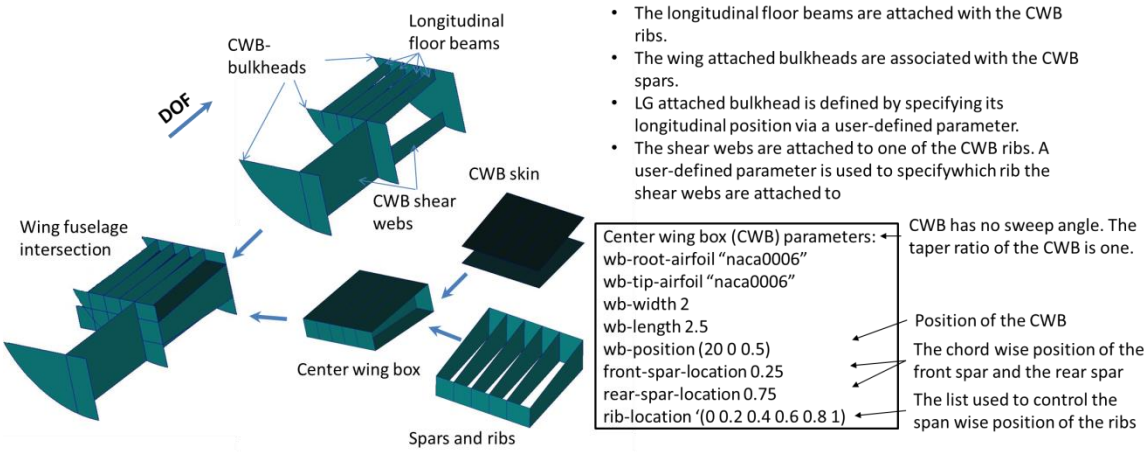


Figure 4.14: Example geometry of a generated wing fuselage intersection

<HLP> Pressure-bulkhead

A pressure bulkhead was modeled as a flat plate in this research to introduce the tension in the fuselage longitudinal direction during the cabin pressurization. The outer boundary of the plate was always a frame line. The inner boundary was a circular curve the radius of which could be specified via input parameters.[1] An example of input parameters for defining a plate representing a pressure bulkhead is shown in Figure 4.15.

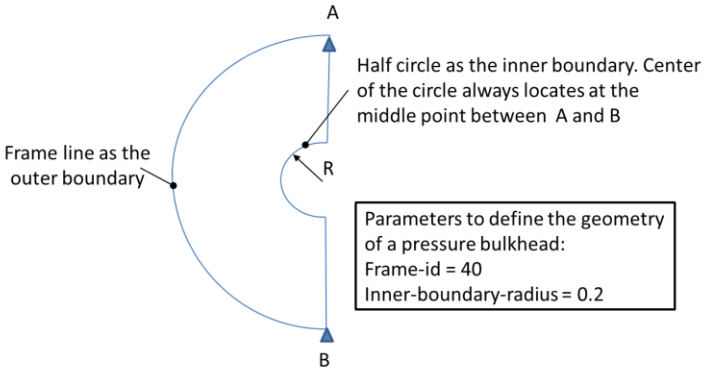


Figure 4.15: Example input parameters for defining a flat surface representing a pressure bulkhead

4.3.2 CMs for load calculation

As discussed in Section 4.3.1, the lifting surfaces of an aircraft are structurally modeled as beams to apply the aerodynamic loads. The *load calculation* calculates aerodynamic loads on all the complete sets of lifting surfaces in trimmed condition under load cases 1-2 and 5. The proper elevator deflection angle is calculated such that the pitch moment about the aircraft CG is zero. Therefore, two input files for AVL should be provided by the aircraft MMG. One is an input file defining the aircraft geometry. The other is an input file defining parameters for AVL running cases, e.g. aircraft weight, and the CG

[1] The inner boundary can be removed if the inner-boundary-radius is specified equal to 0.

position under load cases 1-2 and 5. The process flow for load calculation of an aircraft is illustrated in Figure 4.16.

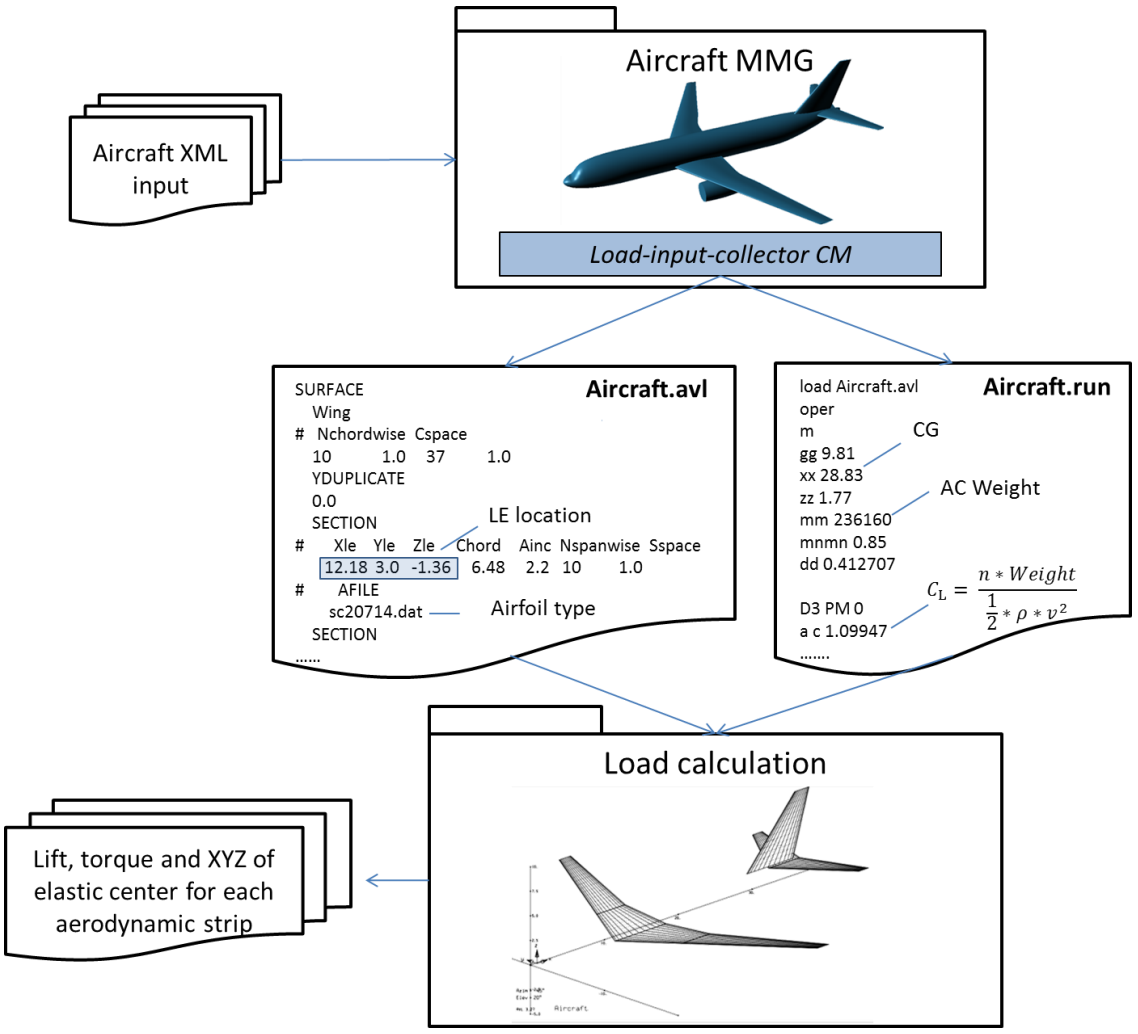


Figure 4.16: Process flow for load calculation

The UML activity diagram to formally describe the load calculation process is shown in Figure 4.17. First, the aircraft MMG loads its input file and instantiates an aircraft model. Then, the *loading-diagram* CM calculates the most forward CG of the entire aircraft. From the aircraft MMG input file the *load-input-collector* CM directly extracts the planform parameters of the lifting surfaces as well as maximum landing weight and maximum take-off weight to write the input files for AVL. The reaction force at the nose landing gear is calculated according to Eq 4.1-4.2. This force is required for analyzing the fuselage FE model under load cases 5-6, see in Table 4.1. As mentioned in Section 4.3.1, the elastic centers of the beams are **assumed** to be at half chord of the root airfoils of the aerodynamic strips in AVL. To get the position of the elastic centers, the coordinates of the LE and TE point for each aerodynamic strip are extracted after running AVL, as well as the aerodynamic lift and torque. How to apply these loads on the FE model is explained in the next sub-section.

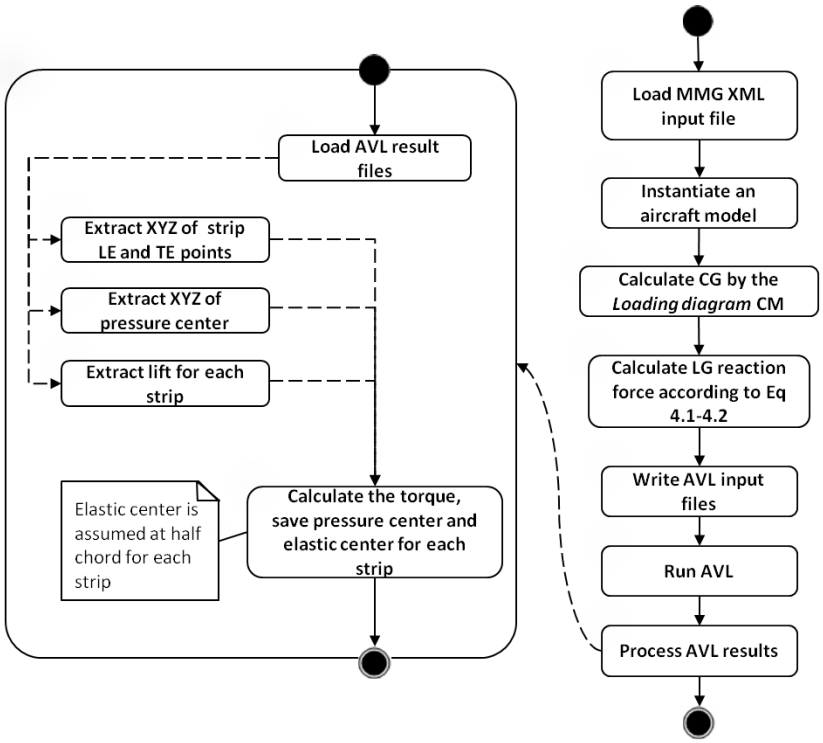


Figure 4.17: UML activity diagram of preparing load data for FEA

4.3.3 CMs for FEA-based sizing

The CMs for the FEA-based sizing method described in Section 4.2.2 are the *ClassII-weight-estimation* CM, the *PCL-writer* and *Panel-sizing-input-collector* CM. These CMs are used to automate the FEA-based sizing process, which has two sequential steps, namely a structural analysis using the FE method and panel sizing. As mentioned in Section 4.2.2, FEA is used to translate external loads such as aerodynamic loads and landing gear reaction forces into internal stresses, whereas the panel sizing method is used to determine the minimum required thickness of fuselage panels.

PCL-writer

As mentioned in Section 3.5, three types of input data have to be collected from the MMG to automate the FE-based structural analysis: the geometric data, i.e. meshable surfaces, the structural properties data and the boundary conditions data.

To get the geometric data, all the skin panels and floor panels are chopped by the cut planes of the *frame* HLP instantiations, which include the virtual frames. These virtual frames are put at the rear fuselage section where the skin panels have a large aspect ratio. Before exporting, the surfaces, which represent structural members of the fuselage, are split into meshable surfaces to ensure surface congruency. For example, the upper and lower skins and ribs of the center wing box are split by the cut planes of the frames between the front and rear spars of the center wing box. After the cutting process, meshable surfaces are exported using STEP or IGES files.

Structural properties data are either directly obtained from the input for the *aircraft MMG* or extracted from the instantiations of the HLPs and other CMs. For example, the cabin systems and equipment are modeled as a non-structural mass smeared onto the shell elements representing fuselage skin panels. These weights are collected from the *Class-II-weight-estimation* CM, whereas the

properties of the floor and floor beams are directly taken from the input. The model details of the fuselage structures and the lifting surface structures are summarized in Table 4.2.

Table 4.2: Structural members modeled by three element types

Group	Element type	Description
Lifting surfaces	CBEAM	Beams used to apply aerodynamics forces
Fuselage skin and stringer	CQUAD	Orthotropic plates with higher axial stiffness considering the stringers' contribution along the fuselage.
Center wing box	CQUAD	Plates used to model ribs, spars and the upper and lower skin. The front spar and rear spar are simply constrained for symmetric load case.
Keel beam	CBEAM	Beams used to reinforce the large cutouts in the fuselage, such as center wing box and landing gear bays
Floor beams	CBEAM	Beams used to transfer the payload to the fuselage stiffened panels; constrain the fuselage deformation during pressurization.
Floor struts	CBEAM	Beams used to transfer loads on the cabin floor to the frames.
Frames	CBEAM	Beams used to transfer the payload forces to the fuselage stiffened panels.
Cabin floor and cargo floor compartment	CQUAD	Plates used to evenly distribute the non-structural mass, such as operational items, cargo containers and payload.
Nose landing gear and tail plane	CONN	Modeled as concentrated masses attached to the main frames. The nose landing gear reaction forces are directly applied on the node to which the nose landing gear is attached.

The boundary conditions data are directly specified via the input of the *aircraft MMG*. All the surface edges on the symmetric plane are applied with the symmetry restraints, red lines in Figure 4.18. Two points, triangles in Figure 4.18, at the front spar of the center wing box are fixed in six freedoms.

The lifting surfaces of the aircraft are structurally modeled to apply the aerodynamic loads as beams, as discussed in Section 4.3.1. The location of the beams' ends is extracted from the load calculation result file while the section parameters, section height and width, are user-defined. The lift and torque on each aerodynamic strip are directly exerted at the beam's end. The process flow of applying the aerodynamic loads on the aircraft lifting surfaces is shown in Figure 4.19. Considering the load transferred from wing to fuselage, the rigid body element shown in Figure 4.20 is used to assemble the wing with a center wing box.

After the three types of data have been collected, the *PCL-writer* CM generates PCL files which automate the structural analysis of the fuselage FE model in PATRAN. The automated pre-processing process for the fuselage FE model is shown in Figure 4.21.

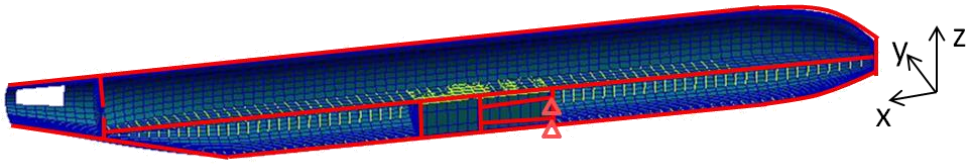


Figure 4.18: Boundary conditions of the fuselage FE model

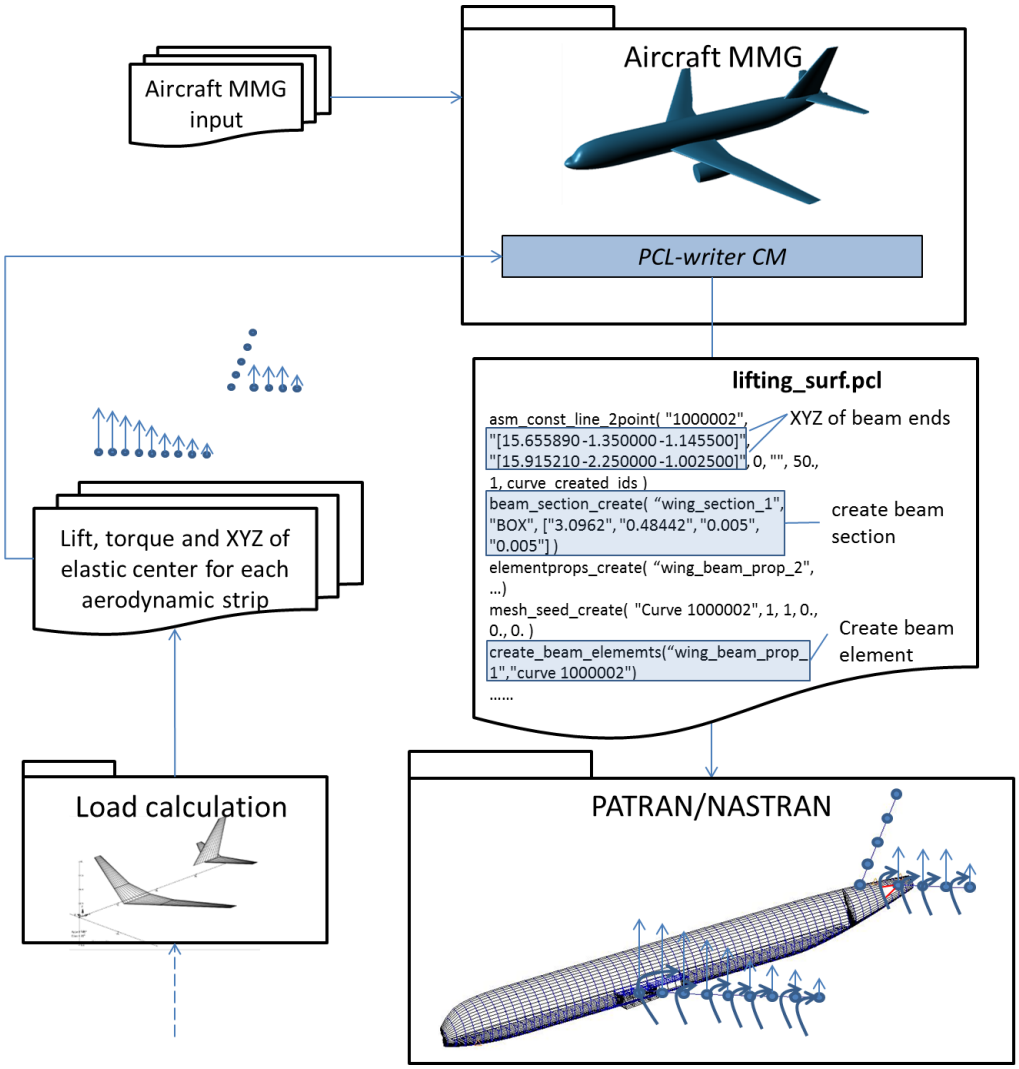


Figure 4.19: Process flow of applying aerodynamic loads on the lifting surfaces. The process flow starts from the load calculation module

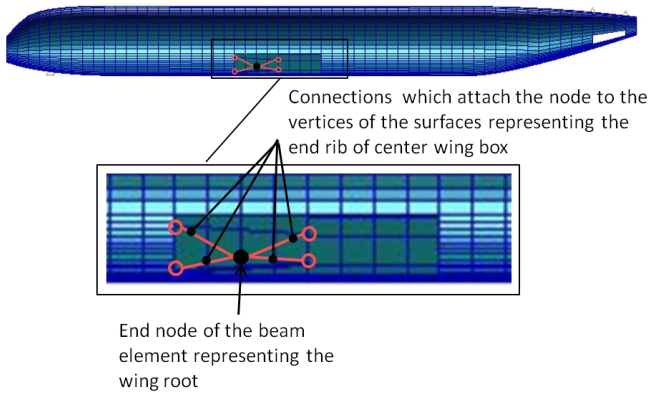


Figure 4.20: A rigid body element used for assembling wing and fuselage

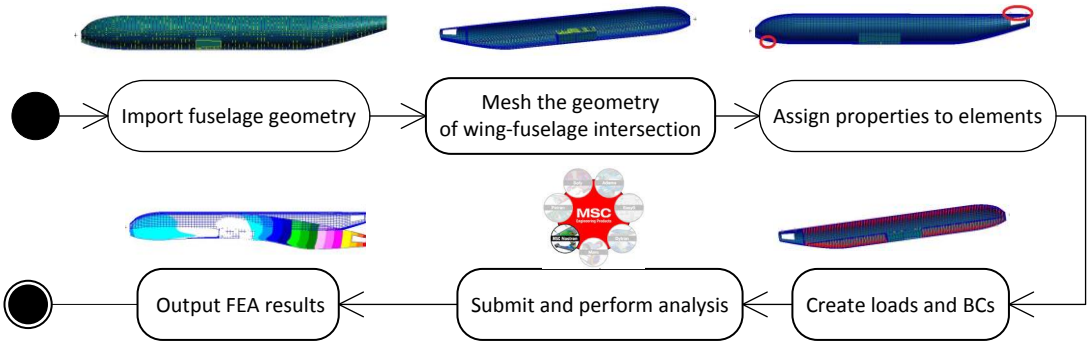


Figure 4.21: Automated FEA for a fuselage

Panel-sizing-input-collector

After the FE model is analyzed and FEA results are generated, the *Panel-sizing-input-collector* CM extracts data from the FEA results, e.g. the longitudinal stress, circumferential stress and shear stress of all the panel elements modeled in PATRAN. All of these data are required for sizing the fuselage panels.

4.4 Application cases and discussion

4.4.1 Conventional aircraft fuselages

The weight estimation using the fuselage ADEE are validated using the actual fuselage weight of various aircrafts from different OEMs: ATR 42, high wing configuration, Fokker 100, T tail configuration, Boeing 737-200, Airbus A320-200 and Airbus A300B2. An example of the generated geometry, the FE model and the vortex lattice model for an A320-200 is shown in Figure 4.22. The required data for the validation, that is geometrical and structural data for the various aircraft, were gathered from literature (Dalen, 1996; Torenbeek, 1982; Obert, 2009) or from specification parameters released by OEMs. The material properties and material strength of the aircraft are shown in Table 4.3. The frame pitch was assumed to be equal to 600 millimeters for all the aircrafts considered since

no exact information was available on the frame position. The error of the prediction was calculated as follows:

$$error = \frac{W_{actual} - W_{calc}}{W_{actual}} \tag{4.14}$$

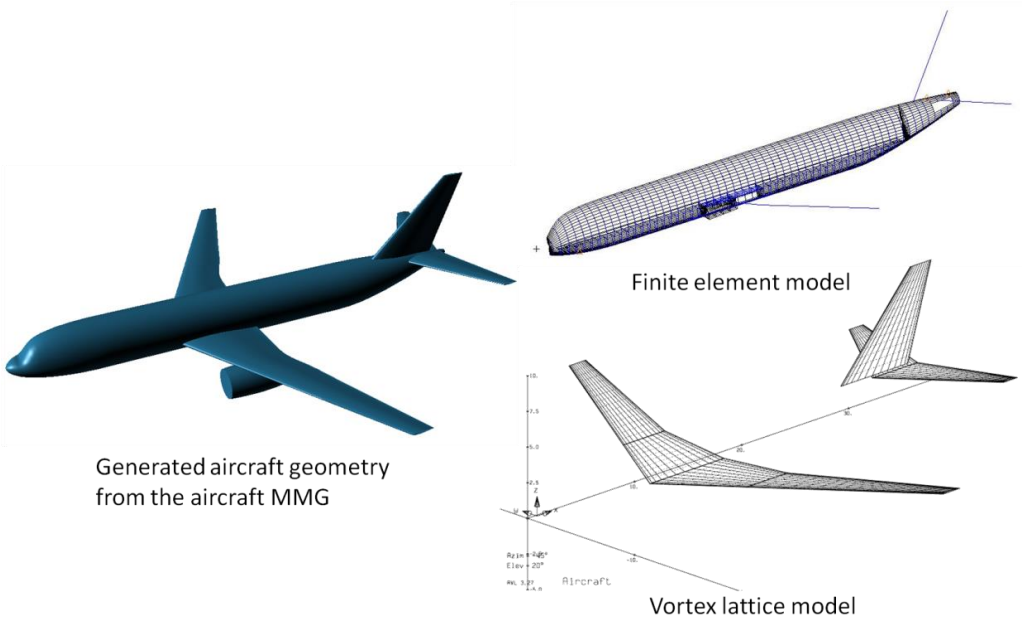


Figure 4.22: The generated geometry, the FE model and the vortex lattice model of Airbus A320-200

Table 4.3: Material properties and material strength. Adopted from van Dalen (1996)

Material	Density (kg/m^3)	E (Mpa)	ν	$\sigma_{\text{res circum}}$ (Mpa)	$\sigma_{\text{res long}}$ (Mpa)	σ_{hoopmax} (Mpa)	σ_{yield} (Mpa)
AL2024	$2.7x10^3$	$72.4x10^9$	0.27	180	69	95	321
GLARE4	$2.39x10^3$	$56.8x10^9$	0.32	200	87	110	321(in metal)

Using the fuselage ADEE, the average accuracy of the fuselage weight estimation was between 5-10% as shown in Table 4.4. From Table 4.5, it can be seen that the weight estimation method used by the fuselage ADEE yields better accuracy than the Torenbeek Class II weight estimation method (Torenbeek, 1982) and the Torenbeek Class II weight estimation method V2 (Torenbeek, 2013) [1]. The skin thickness from the FEA-based sizing is shown in Figure 4.23. Both the weight estimation method of the fuselage ADEE and the Class II method tend to underestimate the fuselage weight of the A320-200. The relatively large error is mainly because the A320-200 has a higher ratio of fuselage faring area to fuselage wetted area than other aircraft, however, the empirical equation for the faring weight estimation is not sensitive to this area ratio. Another reason might be that the A320-200 has a longer

[1] Torenbeek has modified the empirical equation for the fuselage weight estimation in his new book, Advanced Aircraft Design: Conceptual Design, Analysis and Optimization of Subsonic Civil Airplanes, 2013, these equations can be found in Appendix D.

operation service life than the other aircraft while the same fatigue allowable stress was assumed for all the aircraft considered. This requires the A320-200 to have a thicker skin and a heavier fuselage.

It is interesting to find that the level of accuracy of the predicted weight of the aircraft considered is independent of the manufacturers of the aircraft, and no specific tuning of semi-empirical coefficients was used. It is also noteworthy that the weight predicted using the semi-empirical equations took account around 50% of the total structural weight. The level of accuracy of these semi-empirical equations could be further improved by using more aircraft data for a regression study.

Table 4.4: Validation cases for fuselage weight estimation

Aircraft	MTOW [kg]	Actual fuselage weight [kg]	Estimated skin and stringers weight [kg]	Estimated gross shell weight [kg]	$\frac{Gross\ shell\ weight}{fuselage\ total\ weight}$ [%]	Fuselage estimation error [%]
ATR 42	16150	2587	925	1129	43.6	-1.6%
Fokker 100	45850	4758	2003	2493	52.4	3.3%
Boeing 737-200	45540	5277	2091	2605	49.4	2.9%
Airbus A320-200	73500	8938	3150	3967	44.4	-8.1%
Airbus A300B2	136985	16502	7223	9309	56.4	2.5%

Note: Gross shell weight is the total weight of the skin, stringers and frames. The weights of the skin and stringers are calculated based on FEA-based sizing, whereas the frame weight is estimated using Eq D.3 in Appendix D.

Table 4.5: Result comparison of the Torenbeek Class II and Class II V2 method

Aircraft	MTOW [kg]	Actual fuselage weight [kg]	Torenbeek Class II [kg]	Torenbeek Class II V2 [kg]
ATR 42	16150	2587	2155 (-16.7%)	2708 (4.7%)
Fokker 100	45850	4758	5063 (6.4%)	5003 (5.2%)
Boeing 737-200	45540	5277	4594 (-12.9%)	5568 (5.5%)
Airbus A320-200	73500	8938	7473 (-16.4%)	7107 (-20.5%)
Airbus A300B2	136985	16502	18957 (14.9%)	19270 (16.8%)

Note: The fuselage length, width and height used by Torenbeek Class II and Class II V2 are extracted from the aircraft MMG.

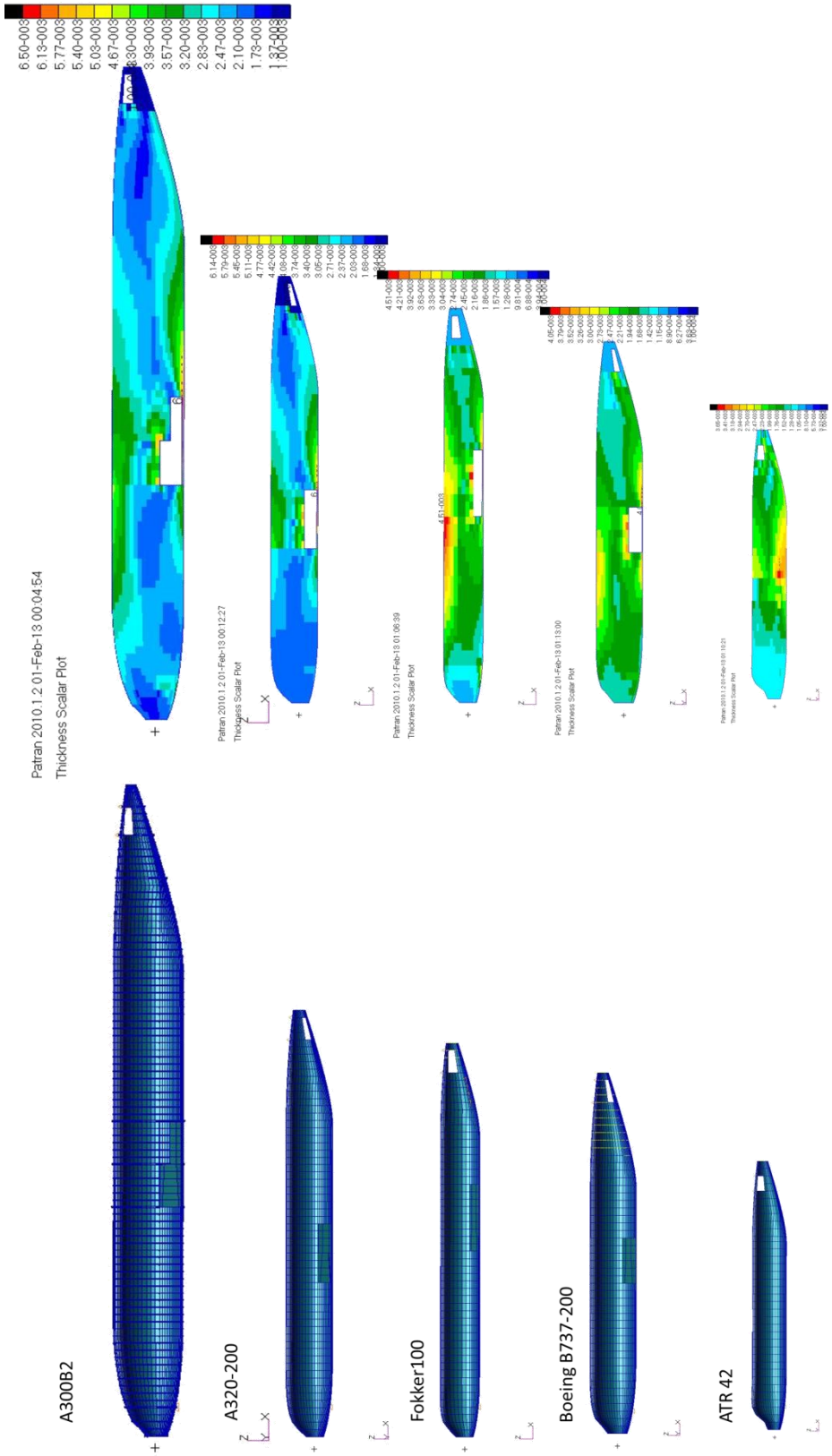


Figure 4.23: Validation cases for conventional configuration aircraft: A300B2, A320-200, Fokker100, B737-200, ATR42, right: FE models; left: resulted skin thickness distribution

Unfortunately, further validation of the presented weight estimation method seems impossible due to the difficulty in obtaining data on the component weights from the aircraft manufacturers, as this is usually highly classified commercial data. Although composite fuselage weight can be estimated using the given material properties and integrating a sizing method for composite skin-stringer panels, the weights obtained cannot be validated since the considered aircraft are from a metallic design before the 1990s.

4.4.2 Prandtl plane fuselage

The empirical weight estimation methods, such as Class II weight estimation, cannot be used for the unconventional aircraft configurations, however, because of its physics-based nature, the method used in the fuselage ADEE can be used to estimate fuselage weight for unconventional configurations, such as a box wing configuration.

The box wing was first studied by Prandtl in 1924 and was found to be a minimum induced drag configuration compared to other wing systems. Although the induced drag might be less using a box wing configuration (Frediani et al., 2005), its weight has to be estimated to understand whether this configuration is structurally efficient. The Class II weight estimation method based on statistics is not reliable for this calculation because the loading on a box wing is different from traditional configurations, as shown in Figure 4.24.

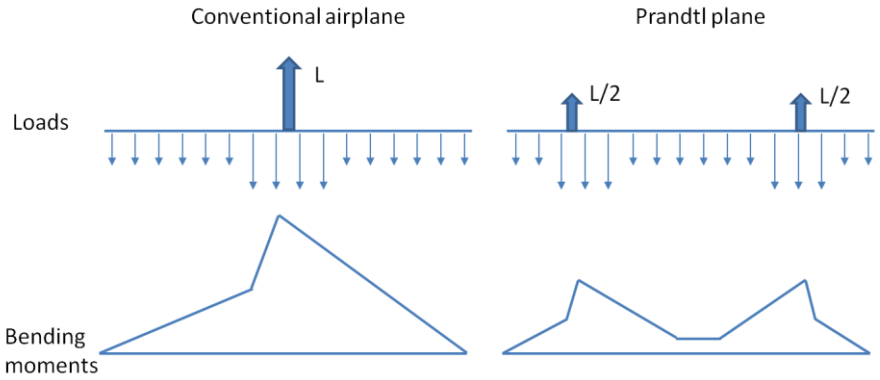


Figure 4.24: Flight loading on a conventional airplane and Prandtl plane

The PraP300 is a baseline design for the Prandtl plane, which is proposed by the University of Pisa and Delft University of Technology. This Prandtl plane can carry 300 passengers with a maximum take-off weight of 236,160 kg (Frediani et al., 2005). The total fuselage length is 51 m and maximum diameter is 5.3 m. The total area of the front wing and the rear wing is 401 m^2, while the sweep of the front and the rear are +35 and -18 degrees respectively. The initial estimated fuselage weight is 21,060 kg. Note: the details of the lifting surfaces are given in Appendix E.

The generated fuselage cabin cross section, the fuselage OML and the FE model are shown in Figure 4.25. The total length of the generated fuselage model is 51 meters while the width is 5.3 meters and height is 4.2 meters. The seat percentages for first, business and economy class are 10%, 25% and 65% respectively.

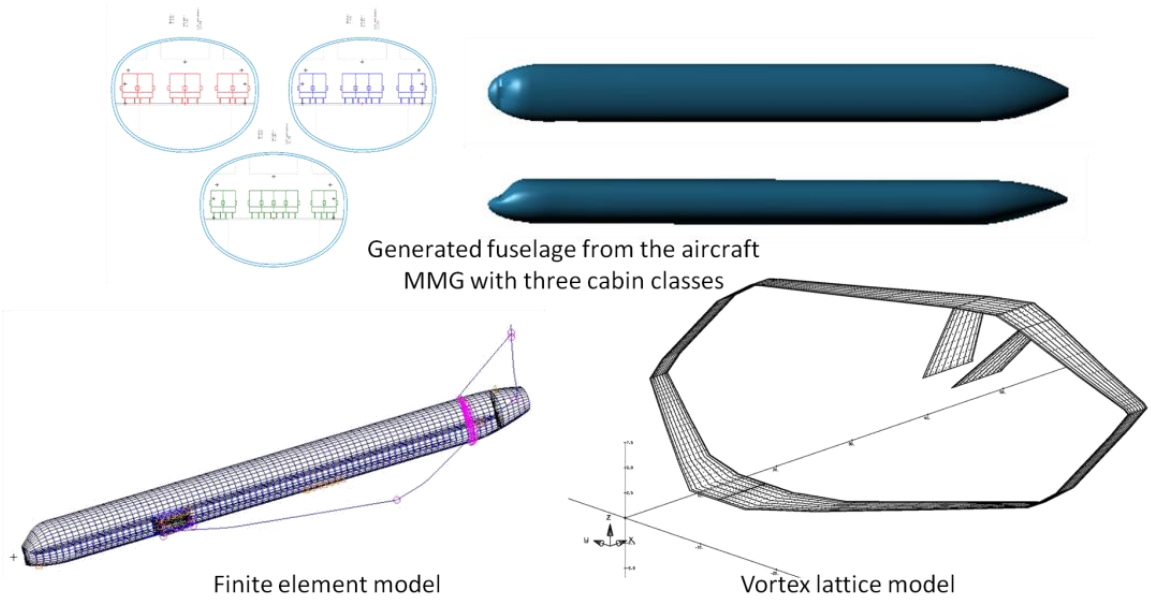


Figure 4.25: The generated geometry, the FE model and the vortex lattice model of the Prandtl plane

For the Prandtl plane fuselage, the estimation result from the Torenbeek Class II method V2 gives a larger deviation from the fuselage weight of the baseline design (see Table 4.6). The predicted weight using the fuselage ADEE was more than 8% of the baseline design. The landing case could easily become the critical load case because the mass of the rear wing, around half of the total wing weight, leads to a larger bending moment along the rear fuselage due to inertia loads during the landing impact.

Table 4.6: Comparison of the estimated weights

Baseline fuselage weight [kg]	Torenbeek Class II [kg]	Torenbeek Class II V2 [kg]	Weight estimation by the fuselage ADEE [kg]
21060	21746 (+3.3%)	13506 (-35.9%)	22744 (+8.0%)

4.4.3 Results and discussion

Although the weight estimation process is automated using the fuselage ADEE, it can be argued that the accuracy of the weight estimation is influenced by the structural layout of an aircraft and the accuracy of the load calculation.

Effect of structural layouts on the weight estimation result

In order to compare different stringer types, the panel efficiency method was used to set the minimum equivalent material thickness of the skin and stringer used in the fuselage to prevent the panel from flexural and local buckling under compression[1]. This method does not require the detailed

[1] It should be noted here that buckling criteria (Eqs. 4.8-4.12) were not activated when this panel efficiency factor method was used to size the skin panels.

description of stringer geometry, therefore, it is suitable to compare the effect of stringer type on the fuselage weight. The method is described in Rothwell (2006), and the panel compressive buckling stress can be calculated:

$$\sigma_{\mathrm{pball}} = \eta \sqrt{\frac{\mathrm{N_x}\, \mathrm{E}}{\mathrm{b_f}}} \tag{4.15}$$

where η is the panel efficiency factor, $\mathrm{N_x}$ is the compressive load intensity, E is Young's modulus and b_f is the frame space (Figure 4.5). The compressive stress within the panel is calculated using

$$\sigma_{\mathrm{pb}} = \frac{\mathrm{N_x}}{\mathrm{t_e}} \tag{4.16}$$

where t_e is the smeared skin thickness, taking into account the stringer thickness.

The design criteria is:

$$\sigma_{\mathrm{pb}} \leq \sigma_{\mathrm{pball}} \tag{4.17}$$

A sensitivity study of the frame pitch b_f and panel efficiency factor η which is a function of stringer type was conducted for the Fokker 100 aircraft. The material property of AL2024 adopted from Dalen (1996) was used and it was kept constant for this parametric study. The Latin Hypercube method was used for 20 design points, 20 combinations of frame pitch and panel efficiency factor. The estimated weight of these 20 designs is listed in Appendix F. The Kriging response surface was generated using MATLAB toolbox DACE, shown in Figure 4.26.

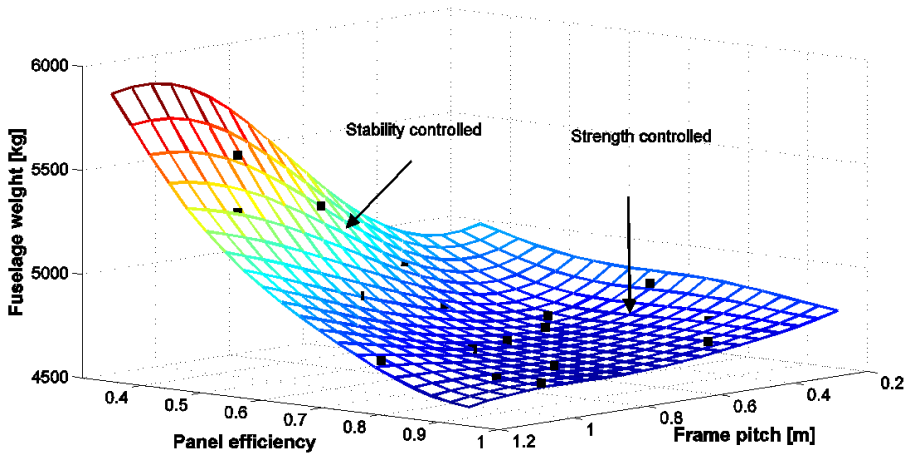


Figure 4.26: The change of the fuselage weight w.r.t. the frame pitch and panel efficiency factor

With regard to the curves on the response surface shown in Figure 4.26, the following could be seen: when fixing the frame pitch the fuselage weight tended to decrease as the panel efficiency factor increased, and this tendency was more obvious with a larger frame pitch. When fixing the panel efficiency factor the fuselage weight tends to increase as the frame pitch increases. The response surface can be used to help the designer to quantify weight savings by varying the panel efficiency factor and the frame pitch.

Effect of different fuselage material

The weight of an A320-200 fuselage with skin and stringers made of GLARE 4 was estimated using the fuselage ADEE. The total fuselage weight yielded 8,023 kg while the skin and stringers were 3,148 kg. Compared with the A320-200 fuselage made from AL2024T3, GLARE gave a 188 kg weight saving, 2.1% of the total fuselage weight.

Effect of load calculation on the weight results

There is no doubt that simplification of the load calculation affects the final weight estimation results. First, the AVL-based load calculation in the fuselage ADEE, for the flight load cases, the maximum positive load factor is fixed at 2.5 g and the minimum negative load factor is fixed at -1 g, however, according to the regulations, both manoeuvre and gust envelopes must be built to take account of all the possible fight conditions that an aircraft might encounter during operation. Second, for the ground and landing load cases, the load factor is also fixed at certain values. Accordingly, asymmetrical loads were not considered in the current load calculation module, which will lead to inaccuracies in the fuselage weight estimation.

These assumed load factor values might significantly affect the estimation results. For example, weight estimation for a Prandtl plane fuselage is sensitive to the load factor value during landing. Increasing the load factor value from 2.5 g to 3.6 g causes the estimated weight to increase from 22,744 kilograms to 26,720 kilograms, adding 17.5% to the total fuselage weight. The 3.6 g load factor was also used to size an unconventional, environment-friendly aircraft configuration (Werner-Westphal et al., 2008). The thickness distribution plots for the two fuselage cases are shown in Figure 4.27.

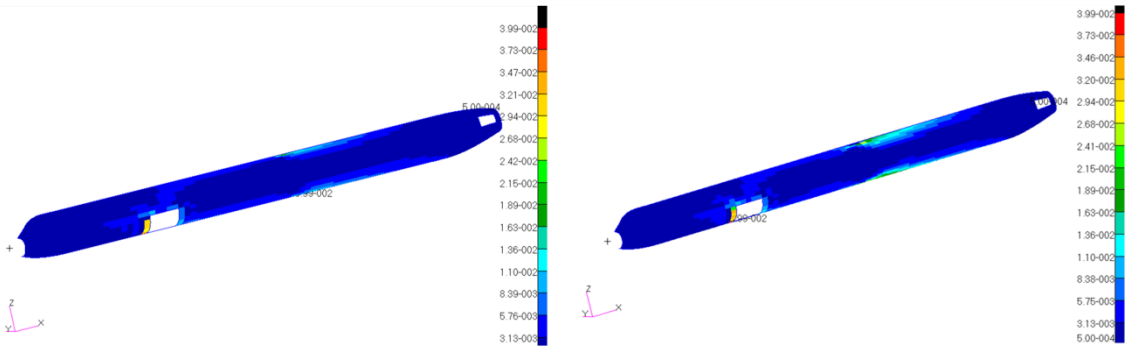


Figure 4.27: Thickness distribution of a Prandtl plane under a 2.5 g (upper) and a 3.6 g (lower) load factor resulting from landing impact

4.5 Concluding remarks

The implementation details of the fuselage ADEE is described in this chapter, and the following conclusions can be drawn.

One, the fuselage ADEE is flexible to handle conventional aircraft fuselages of different sizes, configurations and OEMs. The fuselage weight of an unconventional box wing aircraft can be estimated by the fuselage ADEE.

Two, the estimation accuracy of the FEA-based weight estimation was validated for a number of aircraft fuselages of different sizes and types from different manufacturers. The average weight

estimation accuracy was between 5-10% of that for conventional aircraft fuselages, which is better than results obtained using the Class II weight estimation method.

Three, the weight estimation can be executed with a short lead time by the fuselage ADEE. Due to the automated weight estimation process, the computational time required for each weight estimation, either for a conventional aircraft fuselages or the fuselage of a box wing aircraft, is in the order of 5-10 minutes, using a computer with a 2.00 Ghz Intel Core2Qurd processor and 4Gb RAM memory, and includes the time taken to generate the fuselage structural model, perform the load calculation, determine the panel dimensions, and perform the required iterations.

The combination of computational speed, accuracy and high level design sensitivity seen in the fuselage ADEE makes it a good cross-over which can be used by fuselage panel suppliers to quickly assess the design choices, i.e. frame pitch, stringer type and material strength, on the aircraft weight.

Chapter 5. Implementation and verification of the fuselage panel ADEE

The development of the fuselage panel ADEE, and the use of this tool to demonstrate the global-local knowledge coupling characteristics of high performance is described in this chapter. The types of panels that can be designed contain more details than a fuselage stiffened panel sized by the fuselage ADEE in Chapter 4. The outline of this chapter is as follows: An overview of the panel ADEE is provided in Section 5.1. The structural analysis method, the cost estimation method, and the weight evaluation method used for the panel ADEE are discussed in Section 5.2. The implementation of the panel MMG is described in Section 5.3. A demonstration of automating panel design activities, i.e. panel modeling, FEA-based structural analysis, cost estimation, and weight evaluation using the panel ADEE, is presented in Section 5.4.

5.1 Overview of the fuselage panel ADEE

The panel ADEE was developed to demonstrate that the local design can be accelerated using global-local knowledge coupling. Receiving the fuselage OML and load sets from the fuselage ADEE, the panel ADEE allows the panel designer to perform the panel design quickly, which includes defining a panel concept, analysis of the panel behavior, i.e. cost and weight, and generation of a panel FE model for structural verification. A schematic overview of the fuselage panel ADEE is shown in Figure 5.1. The functionalities of each component within the fuselage panel ADEE are described below.

The *Initiator* is a user interface module, which allows the definition of the inputs related to the product definition and the production definition, i.e. production method for each part and assembly method for each assembly, as required by the panel MMG. The inputs are specified via a Microsoft Excel file.

The *panel MMG* is a panel parametric and generative model which prepares inputs for structural analysis, cost estimation, and weight evaluation tools. As in the case of the panel ADEE, the *panel MMG* comprises HLPs and CMs that are developed in the KBE system. The panel HLPs are developed to generate the geometry of the panel structural members while the CMs generate ready-to-use files for structural analysis, cost estimation and weight evaluation tools. Two CMs are developed for preparing or collecting required data for automated FEA-based structural analysis in Section 5.3: one, a *surface-splitter* for geometry data and two, a *smart-data-collector* for collecting structural properties data. The *PCL-writer* described in Section 3.5 is used to automate the execution of the structural analysis using PATRAN/NASTRAN. The *cost-inputs-collector* CM prepares an input file for the cost estimation tool.

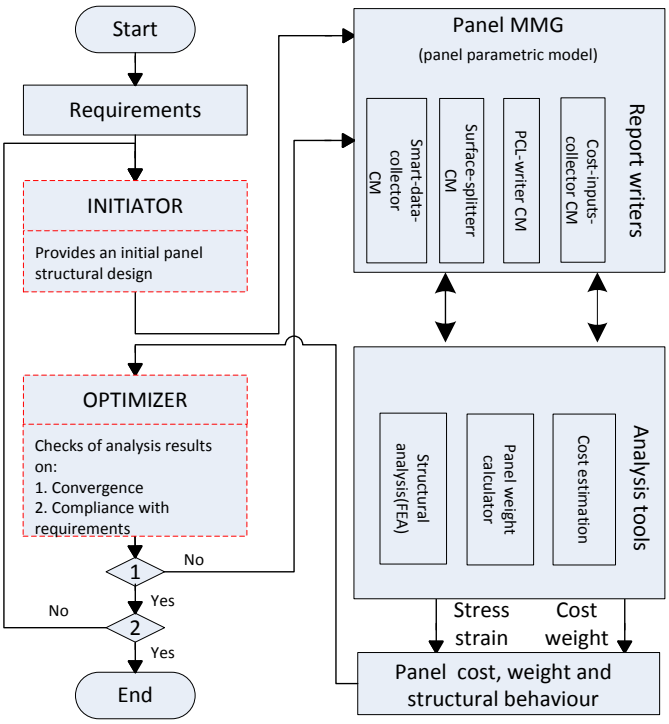


Figure 5.1: Schematic overview of the fuselage panel ADEE, the components in the dashed block are not implemented for this ADEE

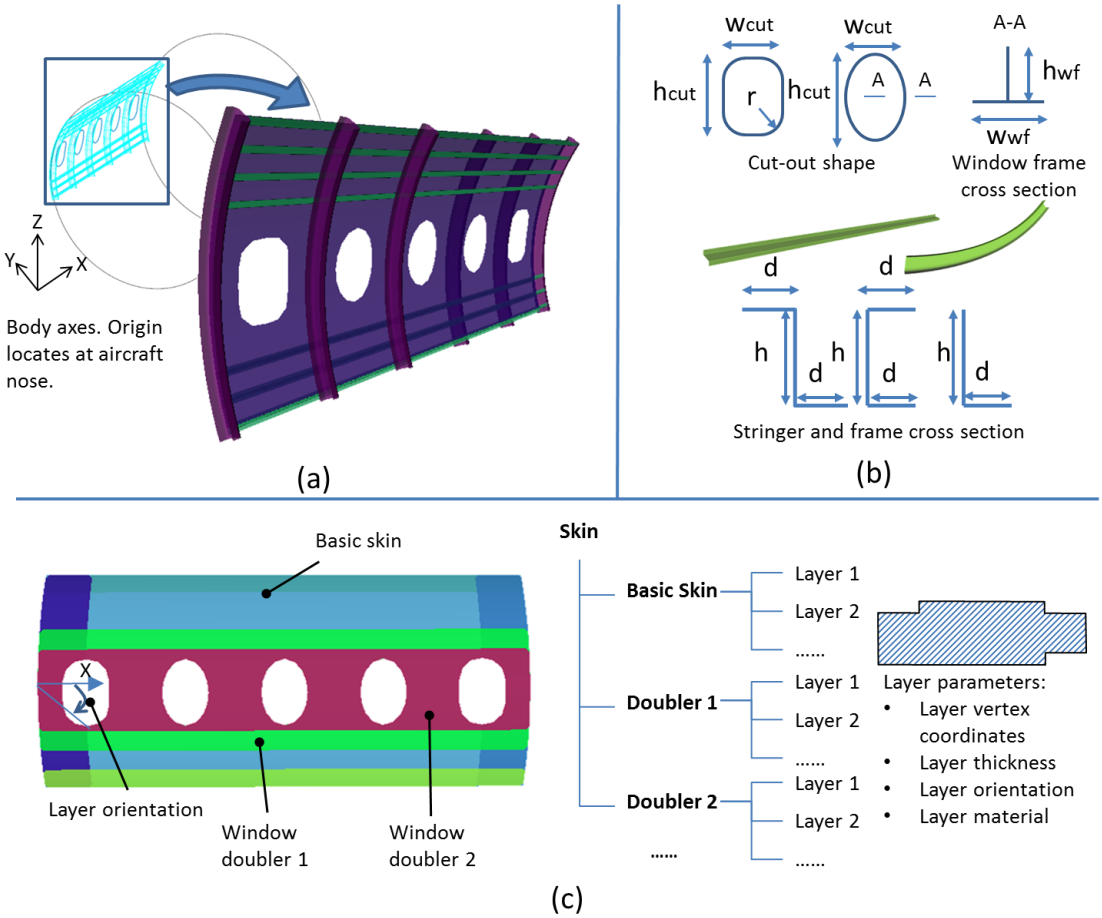
The *Analysis Tools* set consists of PATRAN/NASTRAN for structural analysis, an in-house developed tool for cost estimation, and a weight evaluation tool. The methods employed for structural analysis and cost estimation are discussed in Sections 5.2.1 and 5.2.2. The panel weight is calculated by simply adding all the part weight contributions. The *panel-weight-calculator* is developed within the KBE system to evaluate panel weight. Since the weight estimation tool is developed within the KBE system, it can directly access the product model to obtain the data used for weight calculation. Hence, no specific CMs were developed for weight evaluation.

The *Optimizer* checks whether a panel design meets the structural constraints, i.e. the structural elements do not fail during operation, and or stays within the cost and weight budgets. The focus of this chapter is to demonstrate that the repetitive design actions in local panel design can be automated, the optimization for the panel design will be not addressed in this chapter. The optimization of a local design, such as a movable design, is demonstrated in Chapter 6.

The *Communication framework* was implemented in MATLAB to coordinate the data exchange communication between any two of the components mentioned above.

The fuselage panel considered by the fuselage panel ADEE is the stiffened panel concept, consisting of a multi-layer skin, stringers, window frames and frames. The multi-layer skin is formed from a basic skin and doublers that locally increase thickness, both the basic skin and doublers are further built up by layers, such as metal sheets and fiber reinforced polymer layers with specified material type and orientation. The shape of a layer was chosen to be a polygon, which can be considered to be sufficient to model the layer shape of a skin in preliminary detail. The stringers and frames have three cross section options: Z, C and L. The cut-out shapes implemented in the fuselage ADEE are ellipses and rounded rectangular. Window frames with a T-shape cross section were included in the panel model. Note: see Sub-section 5.3.1 for details about how to position the frames,

stringers and cut-outs, and the definition method for the layer contours. The panel concept with details of the window frame, stringer and frame's cross section and cut-out shapes is illustrated in Figure 5.2.



W_{cut}: cut out width h_{cut}: cut out height W_{wf}: window frame width h_{wf}: window frame height

d_{fr}: frame width h_{fr}: frame height d_{str}: stringer width h_{str}: stringer height

t_{fr}: frame thickness t_{str}: stringer thickness t_{wf}: window frame thickness

Figure 5.2: The fuselage panel design option. (a) The overall fuselage panel structure; (b) details about the shape of cut-outs, and cross sections of stringers, frames and window frames; (c) details about the skin

5.2 Methods for discipline analysis

5.2.1 Structural analysis method

The panel structural analysis was based on the global/local FEA technique which makes use of the following two types of models.

- Global FE model – to capture the overall fuselage behaviors such as nodal displacements and load paths. This is defined by making use of a relatively simplified and coarse FE model, in which the windows and other cut-outs are not included.

- Local FE model – to predict the local fuselage panel behaviors using a detailed and fine FE model. The additional structural features which is included in the local FE model is illustrated in Figure 5.2.

The global FE model is generated by the fuselage ADEE, considering the load cases described in Section 4.2.1. Then, the nodal deformations, displacements and rotations, are applied to the edges of the local panel FE model via a spatial displacement field derived from the global fuselage FEA results. It should be noted here that the element surface loads are applied to the panel FE model for the load cases in which the pressurization is to be modeled. The structural analysis using the global/local FEA that involves the fuselage ADEE and the fuselage panel ADEE is shown in Figure 5.3: because the two FE models are built from the same fuselage OML and the displacement of the global FE model is mapped on the local FE model, the systematic consistency of the geometry and static analysis between these two models is guaranteed.

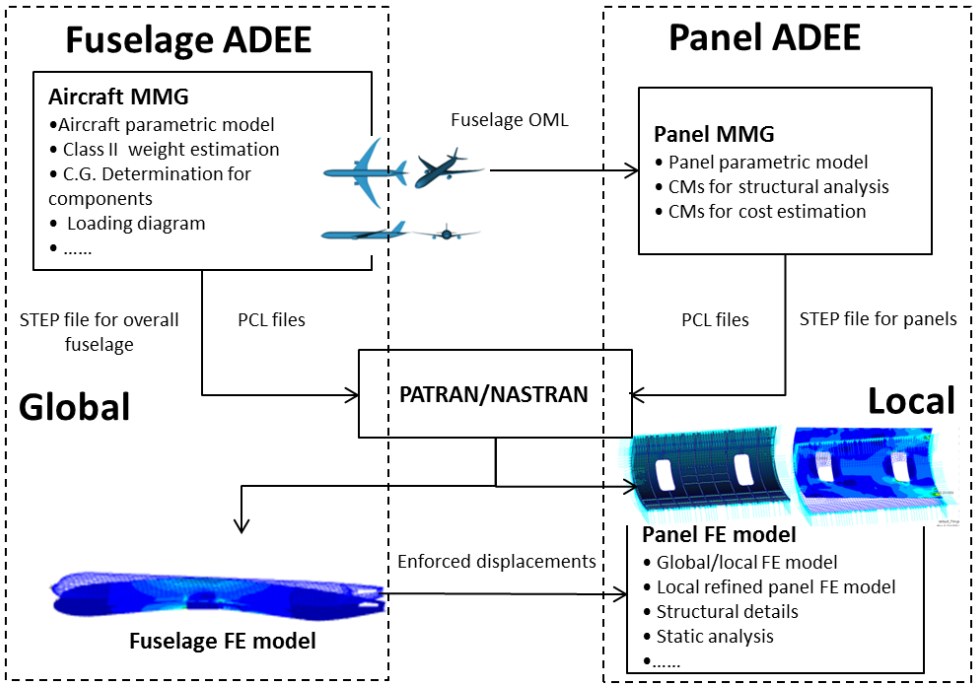


Figure 5.3: Global/local FEA for a fuselage panel

There are two challenges that must be addressed by the fuselage panel ADEE to implement the global/local FEA.

The first challenge rests in the very different level of detail and complexity of the global fuselage model, and local panel model. The local FE model is more detailed than the global FE model; therefore, the local model requires more structural details. For example, the laminate skin and cut-outs have to be modeled in the local FE model, whilst these two features are not supported in the global FE model.

The second challenge is that creating the global/local FE models is repetitive and labor intensive. Several steps, similar to those discussed in Section 3.5, are required to preprocess FE models: importing geometry, assigning shell and beam properties to surfaces and edges, meshing surfaces, creating elements and applying loads and boundary conditions (BCs) etc. The global FE model is

automatically generated as described in Chapter 4. Automatic generation of the local panel FE model also needs to be implemented to accelerate the speed of local design.

5.2.2 Cost estimation method

A bottom-up parametric cost estimation method taken from the literature (van der Laan, 2008; Yin and Yu, 2010; Zhao et al., 2012) was chosen to predict the (recurring) cost of a design. The method imposes a breakdown of the total manufacturing cost into part costs and assembly costs, as illustrated in Figure 5.4. The costs for each part and assembly fall into two different categories, recurring costs and non-recurring cost. The non-recurring costs and energy costs are often estimated based on empirical equations regressed from statistical data. It was not possible to estimate these values for the research presented here, due to the great difficulty with obtaining reliable statistical data from industry to support the regression of empirical equations. Therefore, these costs were not taken into account in the cost estimation module.

The labour time needed to fabricate each part of an airframe and assemble it was estimated using the first-order velocity model (Gutowski et al., 1994)

$$t = \sqrt{(\lambda/v_0)^2 + (2\lambda\tau_0/v_0)} \tag{5.1}$$

where v_0 is the steady-state speed of a manufacturing process, τ_0 is the time a manufacturing process takes to reach 63% of the steady state speed, λ is a cost driver which is usually a geometric attribute of a part, such as area, length and volume.

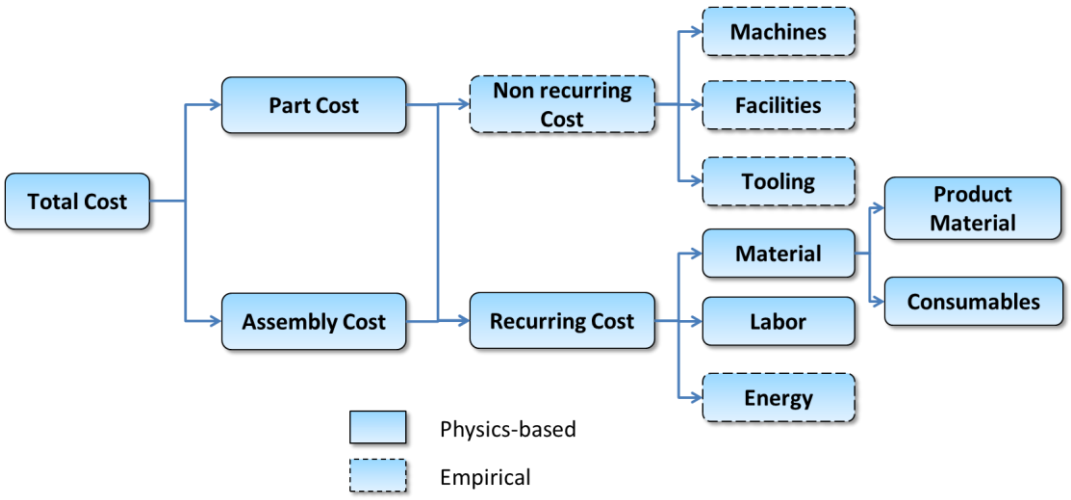


Figure 5.4: Manufacturing cost breakdown

The labour cost is calculated by multiplying the labour time with the labour cost per minute:

$$C^l = t * P_l \tag{5.2}$$

The material used in part fabrication and assembly consists of two types: product materials and consumables. The cost of a material is derived from the amount of the material used in the part/assembly:

$$C^m = x * (1 + sr)P_m \tag{5.3}$$

where x is the governing geometric attribute for material cost estimation, sr is the scrap rate, and P_m is the material price. The thickness and surface area are used as the governing geometric attribute to estimate the material cost of a part, whereas the assembly length/area is used for estimation the assembly material cost.

Given N manufacturable parts and M assembly joints, the total manufacturing cost C^{manu} is given by:

$$C^{manu} = \sum_{i=1}^{N}(C_i^m + C_i^l) + \sum_{j=1}^{M}(C_j^m + C_j^l) \tag{5.4}$$

To perform a cost estimation using this method, a manufacturing concept for the fuselage panels has to be defined first. The definition of a manufacturing concept includes defining the material and manufacturing processes for each part, and defining the assembly method for each assembly. The required data for part cost estimation, i.e. part length, width and area, and the data for assembly cost estimation, i.e. joint length, bonded area and number of clips, must be extracted from the parametric panel model.

5.2.3 Weight evaluation method

The fuselage panel weight is the sum of the weight contributions of each structural member, which is calculated by multiplying material volume and material density. The material volume of beam-like structures, such as stringers, frames and window frames, are calculated by multiplying the cross section area by length. Since the skin consists of several layers, the skin weight is evaluated by adding the weights of all the layers, which are calculated by multiplying the layer volume and layer material density. Layer volume is obtained by multiplying layer contour area and layer thickness. It should be noted that the cut-out area has been taken into account for calculating layer contour area.

5.3 Implementation of the panel MMG

Similarly to the fuselage MMG described in Section 4.3, the panel MMG is built up using high level primitives (HLPs) and capability modules (CMs). The HLPs are *Skin*, *Stringer*, *Frame*, *Window-frame* and *Connection*, which are addressed in more detail in Sub-section 5.3.1. The CMs implemented for preparing inputs for structural analysis and cost estimation are described in Sub-sections 5.3.2 and 5.3.3.

5.3.1 Panel HLPs

The class diagram of the fuselage panel MMG is represented in Figure 5.5. It should be noted that skin and doublers are instantiations of the *Skin*.

Different HLPs may have a similar geometry generation process, and are extended from the same geometry class. The *Stringer* and *Frame* share the *Beam* class, which needs a cross section profile swept along a line. How the *Beam* class instantiates geometry for visualizing stringers and frames is illustrated in Figure 5.6. The red lines are “swept lines”, along which the bottom flange of cross

section profile are positioned. The distance between the cross section centroid and the swept line is half of the height of the cross section web, see Figure 5.6. The cross sections are orientated in a way that they are perpendicular to the fuselage OML surface. This distance, beam offsets, and cross-section orientations are required to assign the beam properties for beam elements in PATRAN.

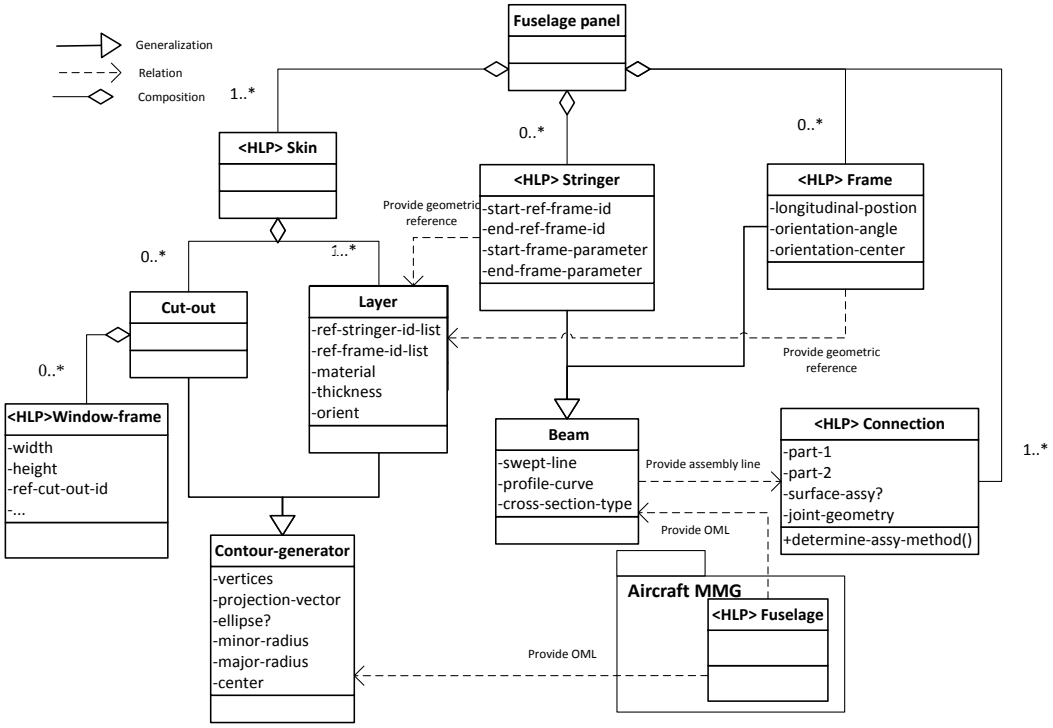


Figure 5.5: UML Class diagram of the fuselage panel MMG

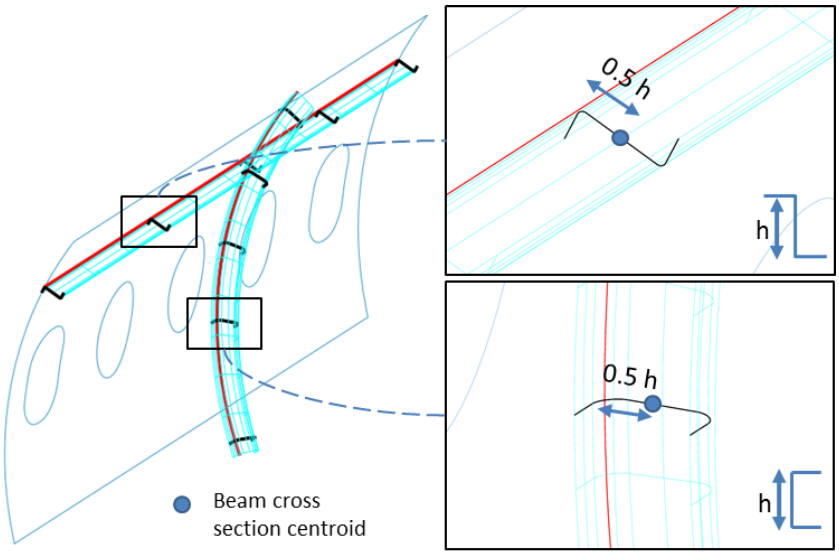


Figure 5.6: Beam class instantiates geometry for visualizing stringers and frames.

The modeling for structural members is introduced of the fuselage panel below:

<HLP>Frame

Similarly to the *Frame* HLP within the fuselage MMG, the *Frame* HLP within the fuselage panel MMG generates a frame cut plane by specifying a plane vector and a point on the plane, Figure 5.7. Then, the reference plane intersects with the fuselage OML and results in an intersection curve. This curve, called the frame line, is passed down to the *Beam* together with the profile curve of the beam cross section. The generation of the frame line is illustrated in Figure 5.7. Finally, the frame geometry is generated by instantiating the *Beam*.

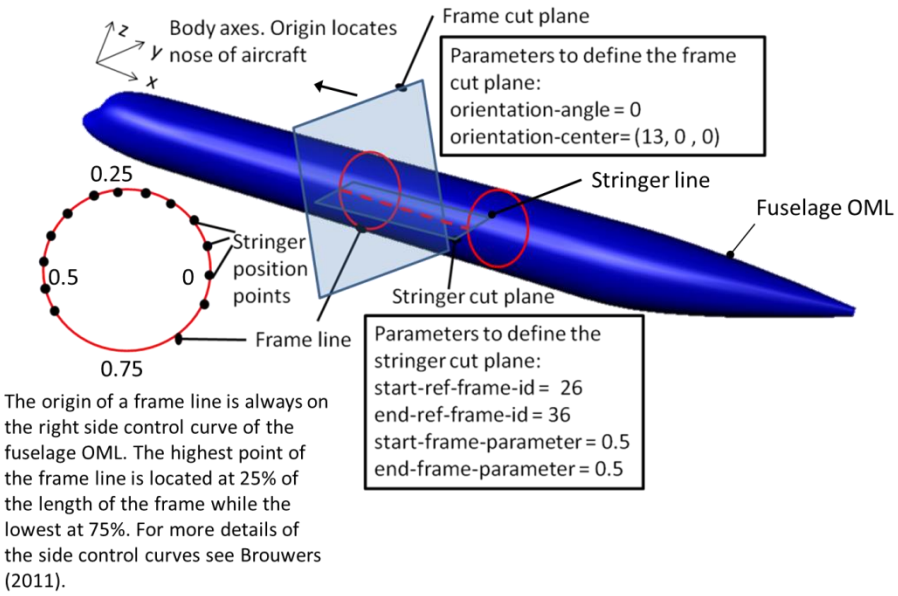


Figure 5.7: Frame and stringer lines definition method

<HLP>Stringer

The generation process of stringer geometry is similar to that of the frame, and the process starts from the generation of the stringer cut plane. First, a line is defined, starting from a point on a frame line and ends at a point on another frame line. Then, a cut plane is built, going through the line. The stringer line results from intersecting the cut plane with the fuselage OML. Finally, the stringer line is passed down to *Beam* to generate the stringer geometry, see Figure 5.7 for an example of the method used to define a frame line and a stringer line. The front frame in Figure 5.7 is located at 13 meters from the fuselage nose along the x axis and the frame cut plane is perpendicular to the x axis. The stringer line, which is represented by the dashed line, starts at 50% of the length of the frame line of the frame 26 and ends at 50% of the length of frame 36 of the rear frame.

<HLP>Skin

The *Skin* HLP is instantiated to represent the basic skin and doublers, which are composed of several layers with polygonal boundaries. Several sequential steps are required to model the geometry of a skin layer. First, each vertex of the skin shape must be positioned by intersecting a frame line and a stringer line. The frame and stringer IDs must be specified. After completing the definition for all the vertices, a surface representing the skin layer is trimmed from the fuselage OML. It should be noted here that the skin layer is represented as a surface without thickness, however, the thickness is added

as a non-geometric attribute (NGA) that links to the geometry representing the skin layer. Considering the presence of cutouts, the surface is further trimmed by the shapes of the cutouts. In an example shown in Figure 5.8, the window doubler contains a single layer. The four vertices (V0-V3) needed to model the layer are defined as the intersection points between Frame 0 and Stringer 4, Frame 0 and Stringer 5, Frame 5 and Stringer 5, and Frame 5 and Stringer 4, respectively. If the positions of the reference frames or stringers change, the vertices will be automatically recomputed, and consequently, the contour of the window doubler layer will adapt. Examples of generated layer shapes are shown in Figure 5.9.

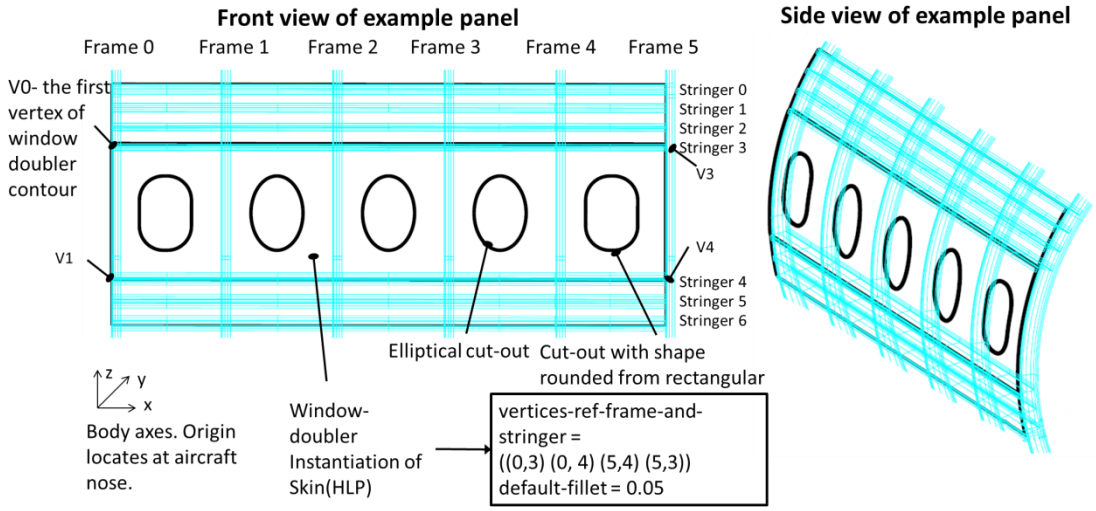


Figure 5.8: Example of vertex definition to extract a layer from a fuselage OML

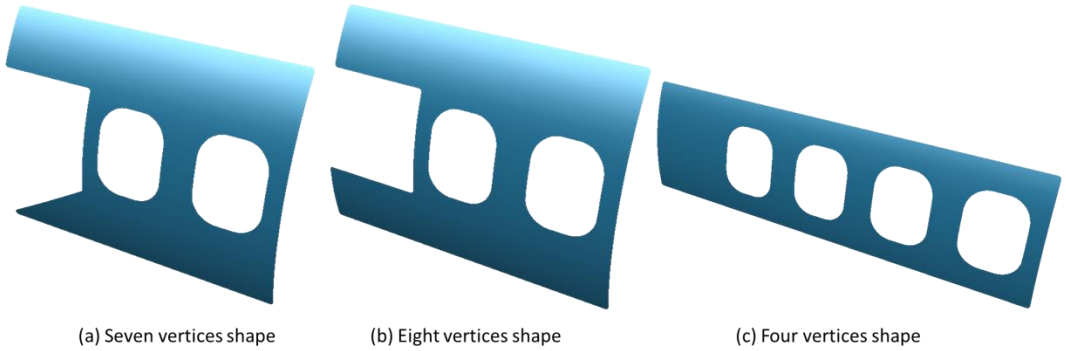


Figure 5.9: Example contour surfaces of layers trimmed from a fuselage OML

<HLP>Window-frame

The *Cutout* class is used to generate fuselage cutouts, e.g. windows and door cutouts. Two different definition methods are used for cutouts. One is similar to the definition method of a skin layer: first the panel designer specifies all the vertices of a polygon (rectangular) shape and then KBE system trims the shape from the fuselage OML. The other method is used to define an elliptical cutout which requires two steps are required for defining such cutouts. One, an elliptical curve is constructed on the XoZ or XoY plane by specifying its center, major and minor radius. Two, the elliptical curve is projected on the fuselage OML and the cutout shape results. After defining the shapes for all the

cutouts, the *window-frame* HLP uses the boundary curve of a specified cutout as a geometric representation of a window frame with a T-shape cross section. An example of window frames is illustrated in Figure 5.10. A web height and flange width are required to assign beam properties in PATRAN.

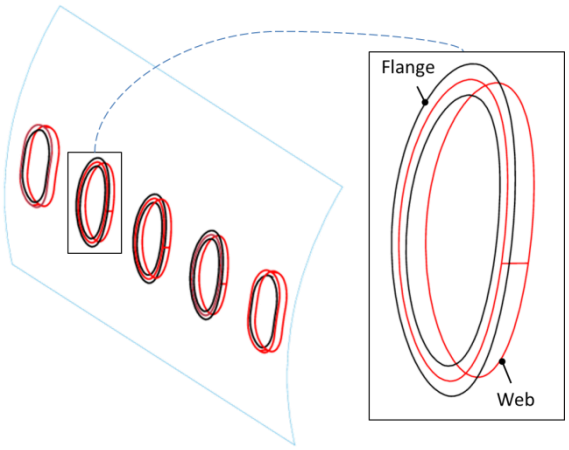


Figure 5.10: Example of window frames

<HLP>Connection

The *Connection* HLP is instantiated to represent a joint between two parts. A joint is geometrically represented by an overlapped surface of two parts when doublers are assembled to basic skin. When a beam-like structural member such as a stringer/frame is assembled with the panel skin, the swept line of the structural member is extracted to geometrically represent the joint. In a case where a window-frame the cut-out boundary curve is extracted. In the case that two beam-like structural members are assembled together, the intersection point of the swept lines of these structural members is used. For example, the clip that connects a stringer to a frame is represented geometrically as a point. The above three types of geometrical representation using this HLP are illustrated in Figure 5.11.

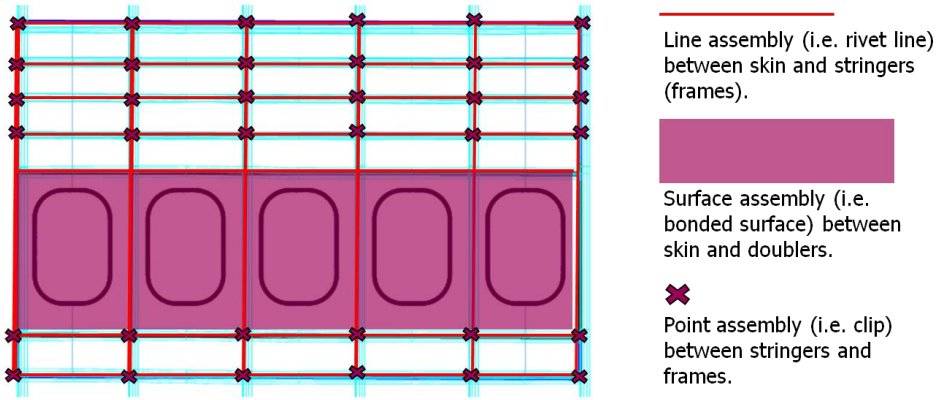


Figure 5.11: Three geometric representation for connections: lines, surfaces and points

"Virtual elements" are introduced in the panel MMG to provide a positioning reference to define a layer or window frame, or to support the cutting process of the skin surface into ready-to-mesh surfaces. The cutting is implemented by the *Surface-splitter* CM which isintroduced in the next

section. Note: These virtual elements are not accounted for in the cost estimation and weight evaluation, nor included in the local FE model.

The process of defining a panel structural configuration is shown in the activity diagram presented in Figure 5.12. The complete set of input parameters for the panel MMG is given in Appendix G. From Figure 5.13 it can be seen that different panel configurations can be modeled from different fuselages, and in Figure 5.14 an example panel that includes the frames is shown.

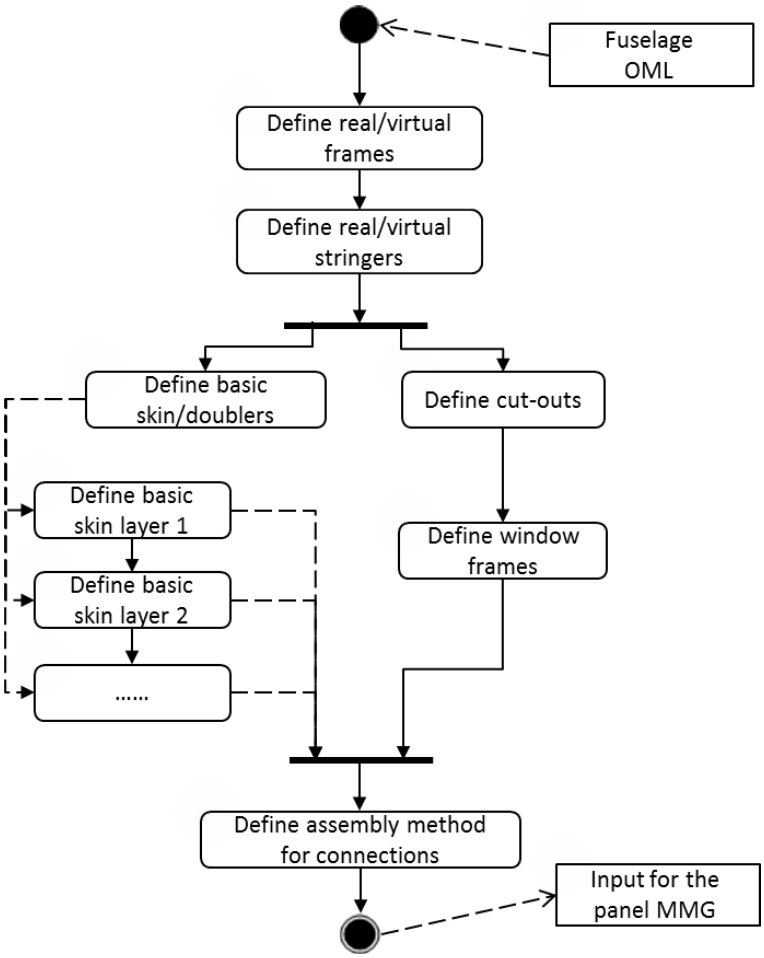


Figure 5.12: Activity diagram for defining a panel structural configuration

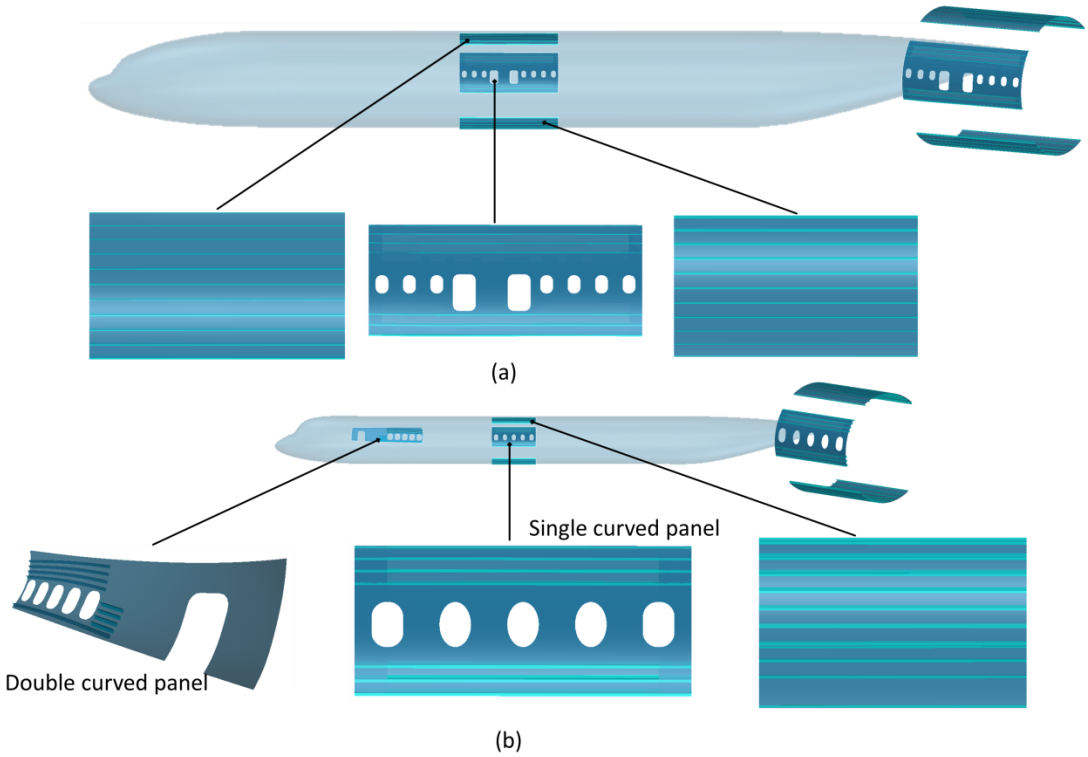


Figure 5.13: (a) panels in a large commercial aircraft and (b) panels in a regional aircraft

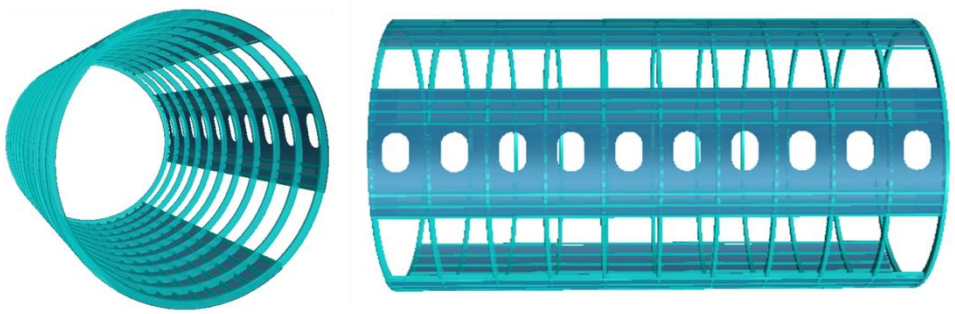


Figure 5.14: An example of panels with frames

5.3.2 CMs for structural analysis

As discussed in Section 3.5, three types of data are needed for a structural analysis: geometry data, structural properties data and boundary conditions data. The *Surface-splitter* CM is used to generate ready-to-mesh surfaces from which the discretized FE model of fuselage panel is constructed. The *Smart-data-collector* CM collects the structural properties data, e.g. thickness, layer orientations and layer material, for composite/monolithic fuselage skin and beam-like structural members, such as frames, stringers and window frames. The boundary conditions data, such as nodal displacements of panel edges, are provided by the global fuselage FE model, and hence this is not be handled by the CMs of the panel ADEE. The *PCL-writer* is used to generate the PCL files that automate the entire pre/post-processing process in PATRAN. Details of these CMs are provided below.

Surface-splitter

In the panel parametric model, the KBE system generates the surfaces of the skin and doubler. These surfaces are generally not ready to be exported into PATRAN because:

- different surfaces have different properties. As shown in Figure 5.15, the skin is built up out of three laminates, basic skin, doubler 1 and doubler 2, which are modeled in the KBE system by instantiating the *Skin* HLP. Each S*kin* instantiation comprises several layers with user-defined thickness, material and orientation. Part of the skin surface is covered by doublers while the other part is not, leading to different numbers of layers and stacking sequences. The skin surface must be split into smaller surfaces, which will be individually assigned with different properties in PATRAN.
- the structural engineer needs to check the stress values for the skin within each stringer bay. This means the skin surface has to be split at least by the frames and stringers.

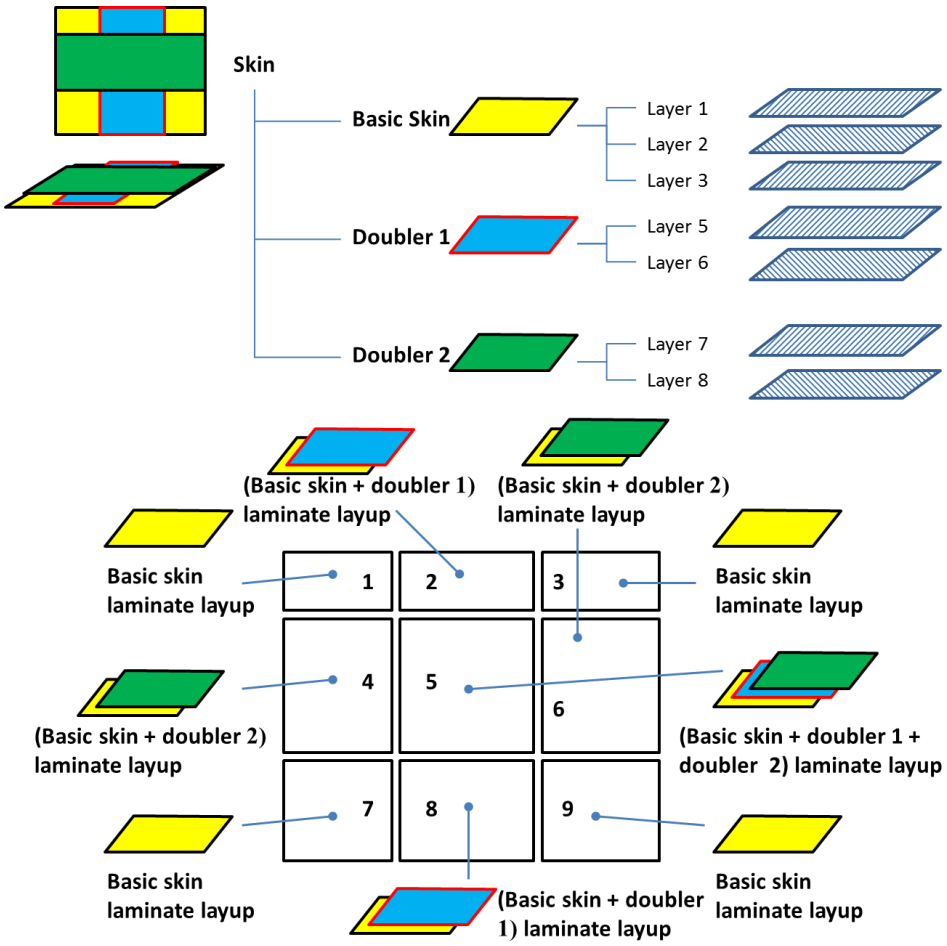


Figure 5.15: A composite skin surface is divided into segments with different numbers of layers and stacking sequences.

Remark: the contour surface of a skin layer may cover several segments, for example, the contour surface of Doubler 2 covers segments 4-6, see Figure 5.15.

At the beginning of the surface splitting process, the *surface-splitter* scans the entire product model and collects all the reference planes of the frames and stringers, including the virtual ones, then it cuts the skin surface using these planes, and collects these segments into a *segments-list*. For traceability purposes, each segment is assigned a unique number, which indicates its location inside the *segments-list*. Finally the *surface-splitter* generates a STEP file which contains all the segments.

Smart-data-collector

Different segments can have different properties, segments 1-9 in the panel model in Figure 5.15 have different thicknesses and laminate layups. The right properties have to be assigned to the right segments. The *smart-data-collector* collects the property data for each segment required by the *PCL-writer* to generate PCL files which are used to create shell elements in PATRAN. The two important types of data collected for each segment from the panel product model are:

- *Mapping-list*[2]. This list is used to indicate whether a given segment is covered by the contour of a layer. The n^{th} element of the *mapping-list* is a Boolean value which is used to notify whether the contour of layer n covers the given segment. The *mapping-list* for segment 8 in Figure 5.15, for example, is '(t t t t t t nil nil). This indicates that segment 8 is covered by layers 1-6, and that layers 7-8 do not cover segment 8. In order to get this list, the geometric centroid of the segment has to be found, and the normal vector at the centroid point. Then, the KBE system draws a line that goes through the centroid and is parallel to the normal vector. The line is intersected by the surface of each layer. If an intersection point is found, "t" is added to the *mapping-list*; otherwise "nil" is appended. After the *mapping-list* is obtained, the *smart-data-collector* CM extracts the non-geometrical attributes of layers 1-6 required to assign properties in PATRAN. When the material types of layers 1-6 are not the same, data is extracted from the two layers, i.e. stacking sequence, thickness of each layer, and material of each layer. The workflow of the *mapping-list* generation process is shown on the left side of Figure 5.16.
- Type. This attribute is used to indicate whether the given surface is a monolithic or a composite plate. When multiple *skin-sheet* objects are detected in one segment, the *smart-data-collector* first compares the layer material types. If all the *Layer* objects are made of the same material, the type is set to monolithic; otherwise the segment is modeled as a composite shell in PATRAN.

[2] One *Mapping-list* per segment.

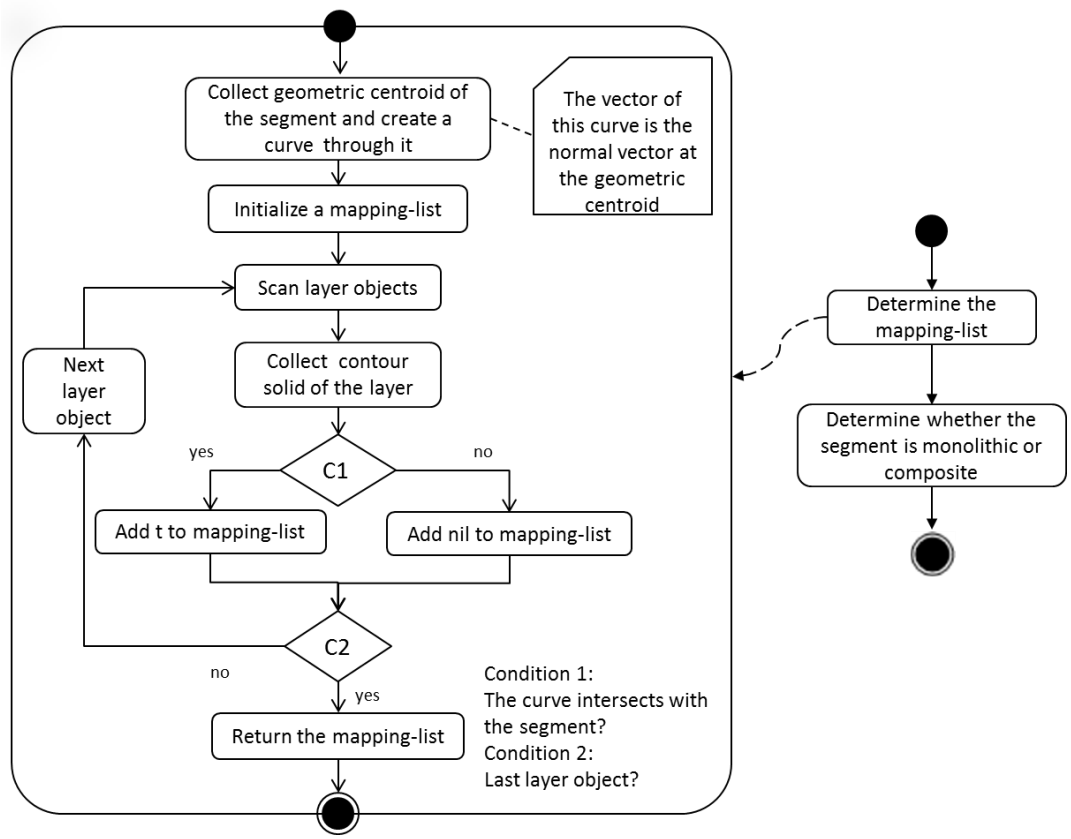


Figure 5.16: Workflow diagram of property data collection process for a segment

The *PCL-writer* needs the structural properties data, i.e. the cross section parameters, beam offsets and materials, to create finite elements for beam-like structural members in PATRAN. These data are described as follows.

- Beam structural properties data: material type, section type, orientations, and dimensions (see Figure 5.2(b)). The orientations of the cross-section are directly extracted from the instantiation of the *Beam*.
- *Segment-edge-id-list* containing the ID of the segment edges to which a beam is attached. The workflow diagram, Figure 5.17 left, illustrates how the *smart-data-collector* gets this list. An example of a *Segment-edge-id-list* is '((23 1) (27 2)), this indicates that the 1st edge of segment 23 and the 2nd edge of segment 27 are modeled as beam elements. In order to obtain the *Segment-edge-id-list*, the swept line of the beam instantiation is first extracted, then, the KBE system starts to scan all the segment edges. If the end points of a segment edge are on the swept line, the segment ID and edge ID are added into the *Segment-edge-id-list*. The properties assigned to the beam elements in PATRAN are extracted from the non-geometrical attributes of the *Frame*/*Stringer*/*window-frame* instantiations, i.e. material, offsets, dimensions and cross section orientation vector.

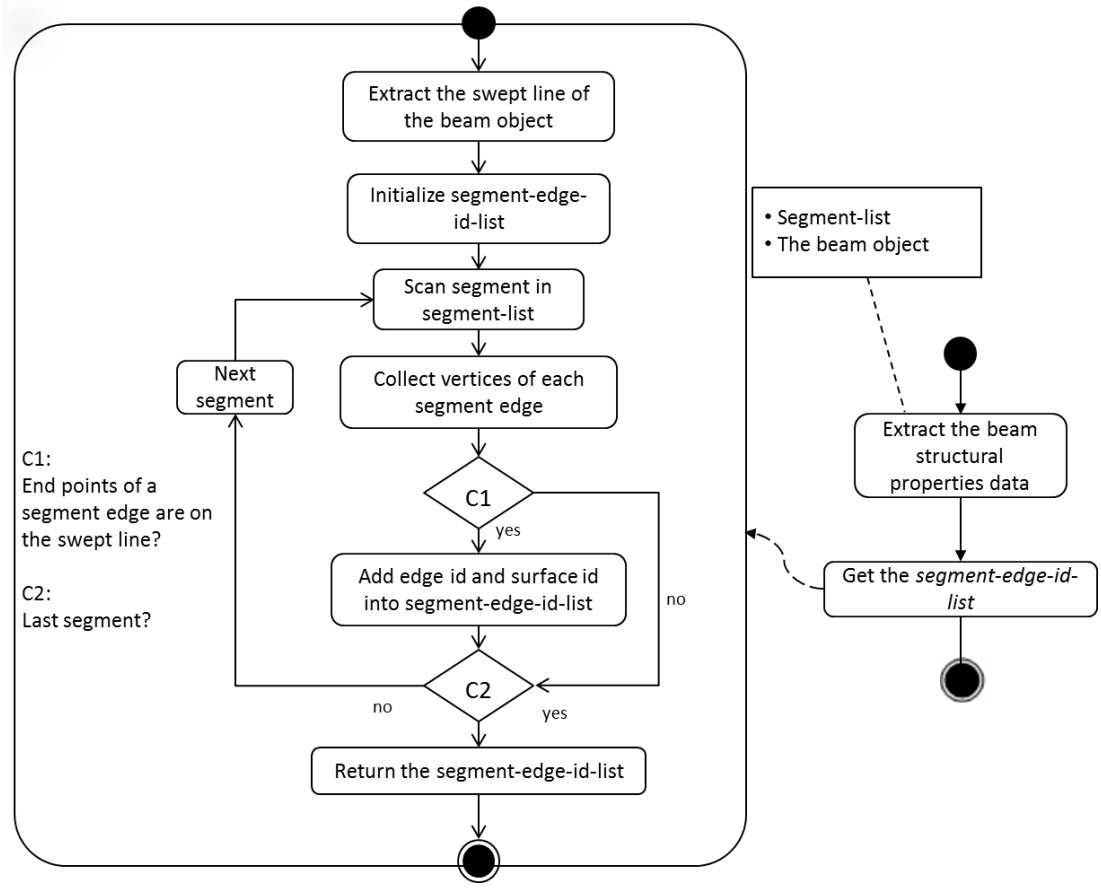


Figure 5.17: Workflow of property data collection process for a beam-like structure

PCL-writer

The PCL files are generated using the data collected by the *smart-data-collector* CM and the geometry data provided by the *surface-splitter* CM. An overview of the automated execution process of the global/local FEA in PATRAN is given in Figure 5.18.

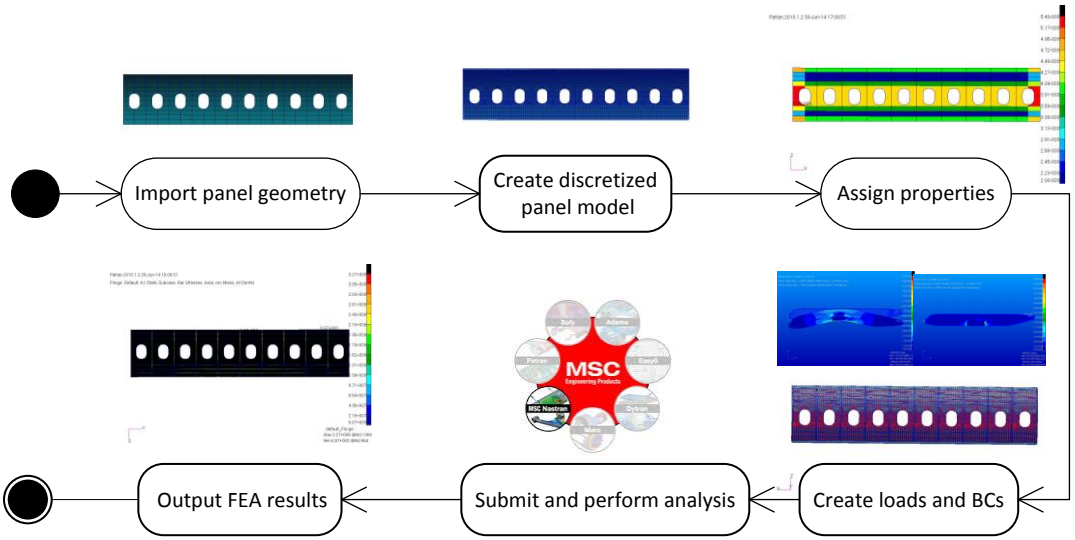


Figure 5.18: An example of automated execution process of the global/local FEA

5.3.3 CMs for cost estimation

The *cost-inputs-collector* prepares the inputs required for the cost estimation shown in Figure 5.19. First, the *cost-inputs-collector* CM scans all the *Skin* HLP instantiations in the parametric model, the part ID, width, length and area are extracted from these instantiations, together with data about the manufacturing processes, then the CM extracts dimensions of cutouts, see Figure 5.2, and the cutting method, e.g. chemical milling for metallic skin. The *cost-inputs-collector* collects the parts' ID, profile thickness, width and length of the stringers and the frames. Finally, the CM collects inputs to calculate the assembly cost. If a connection is geometrically represented by a line, the line length and assembly method is collected. In the case where a connection is geometrically represented by a surface, the surface area and assembly method are collected by the CM. The activity diagram for generating a ready-to-use input file for cost estimation is shown in Figure 5.20.

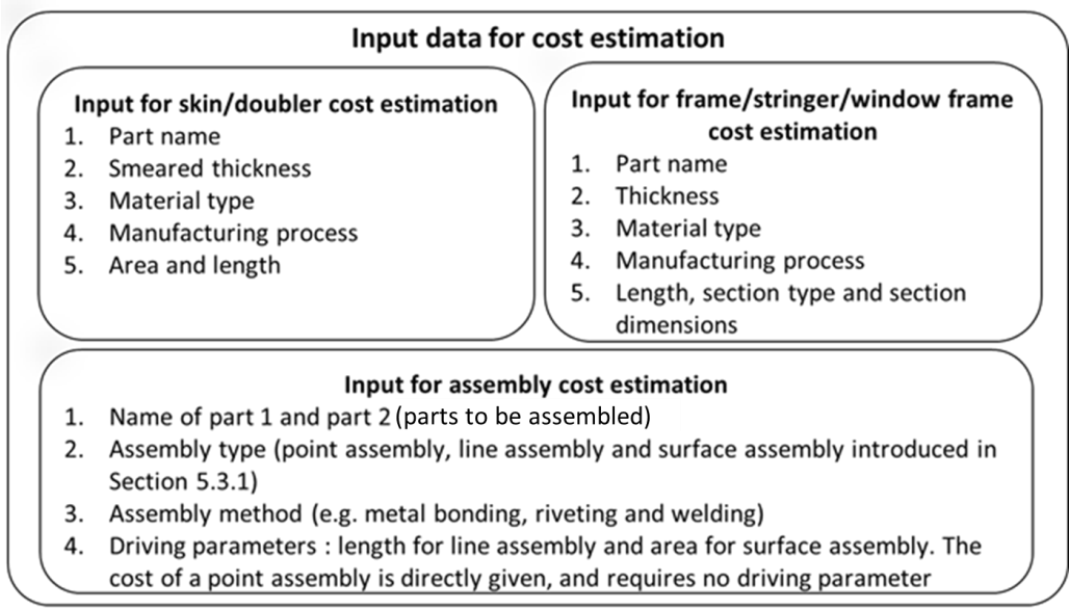


Figure 5.19: Input data required for cost estimation

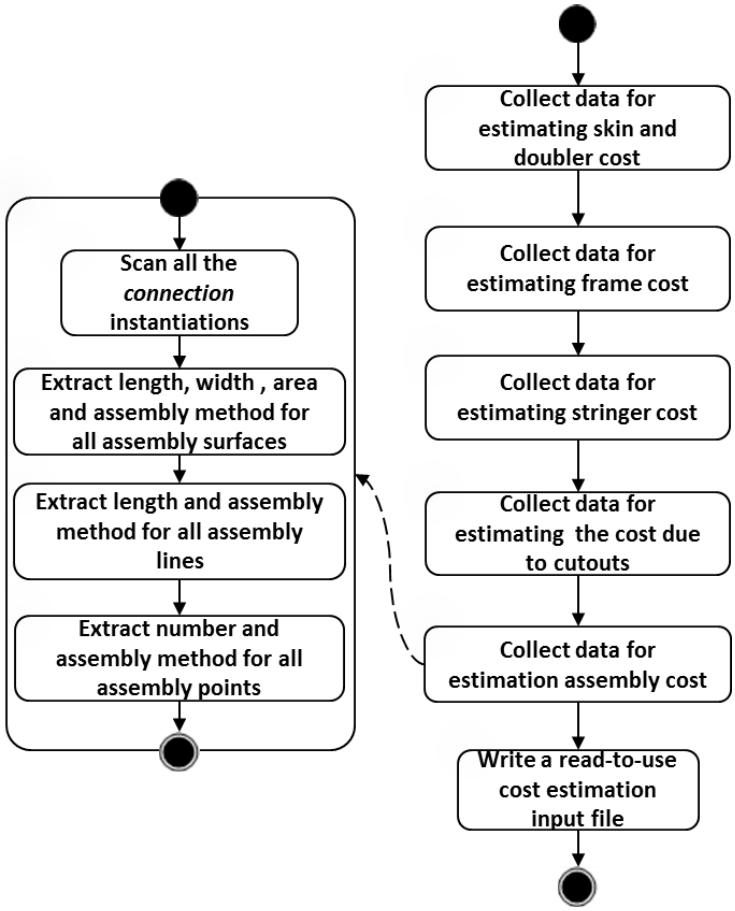


Figure 5.20: Activity diagram for generating a ready-to-use input file for cost estimation

5.4 Demonstration of the fuselage panel ADEE

A fuselage panel from a regional aircraft was chosen to demonstrate the panel ADEE's functionalities. The panel was located 13 m away from the fuselage nose along the fuselage longitudinal direction. The panel length was 2.5 m. The panel was composed of seven machined aluminum stringers, six machined AL7075 frames and a skin bonded by seven AL 2024 sheets. The panel structural layout and the values of the design parameters are illustrated in Figure 5.21. The load case 1, 2.5g + 1 pressure differential, was chosen to analyze the global fuselage model to get the nodal displacements of the panel edges.

After the required inputs were specified via an Excel worksheet, the panel ADEE fully automated modelling the panel, performing the global-local FEA and carrying out the cost estimation and weight evaluation. The tool was run on a computer with a 2.66Ghz Intel Core2Qaud Q8400 processor and 4Gb RAM memory. The calculation time for each process step are depicted in Table 5.1and the total calculation time per configuration is calculated. The total panel weight evaluated by the ADEE was 58.4 kg, whereas the recurring cost was 12093.3$. A detailed local panel FE model was automatically generated, and the nodal displacements of the panel edges calculated from the global FEA. A detailed weight and cost breakdown of the fuselage panel is shown in Figure 5.22. The process, including the automatic generation of the panel FE model, is depicted in Figure 5.23.

Table 5.1: Computational time of the demonstration case

Step	Time
Define panel configuration and manufacturing concept	240 s
Generate panel model and prepare inputs for disciplinary analysis tools	120 s
PASTRAN preprocesses global/local FE model	120 s
NASTRAN perform static analysis	60 s
Cost estimation	30 s
Weight evaluation	30 s
Total time	~10-min

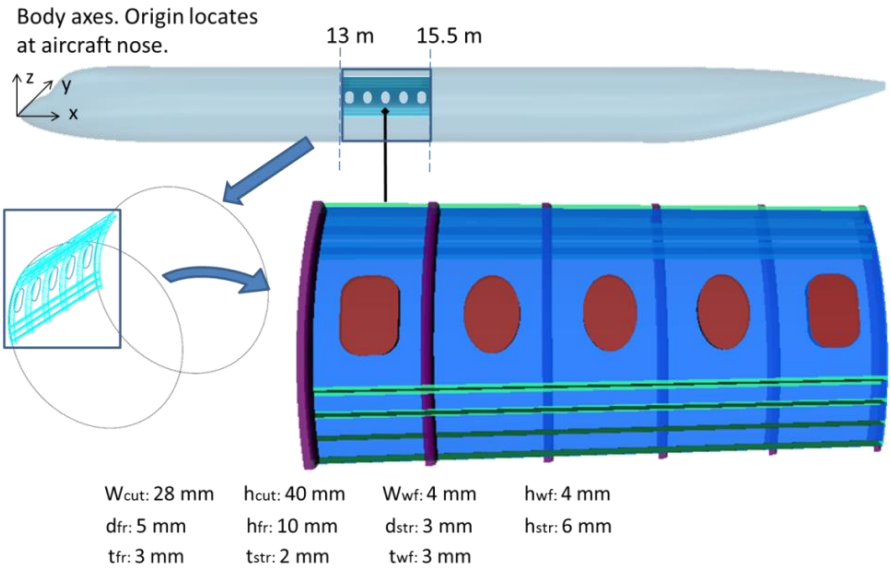


Figure 5.21: The verification panel structures

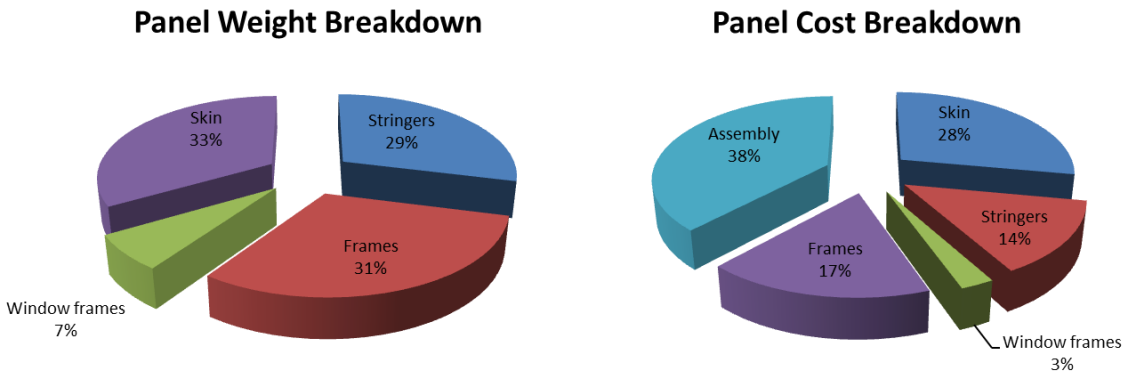


Figure 5.22: Panel weight and cost breakdown.

It was found that the repetitive activities in the panel design process were automated by capturing the process knowledge in the design process. The process knowledge of preparing ready-to-mesh

surfaces was captured in the *surface-splitter*. The process knowledge of collecting the structural properties data for preprocessing the panel FE model was captured by the *smart-data-collector*. The process knowledge for generating the panel FE model was captured by the *PCL-writer*. The process knowledge of preparing inputs for the cost estimation was captured by the *cost-inputs-collector* CM.

5.5 Discussion

Considering the requirement for global-local knowledge coupling, the fuselage ADEE, as the cross-over, provides the necessary inputs for the local panel design: the fuselage OML surface and panel loads for the local panel design. At the local level, a parametric and generative panel model in the panel ADEE is developed to include the geometric features that are required for a bottom-up cost estimation, weight evaluation, and refined FEA of a fuselage panel.

The innovation of the panel ADEE is that the repetitive panel design actions are now automated, i.e. panel modeling, refined panel FEA preprocessing, and cost/weight estimation. Traditionally, the panel modelling and the preprocessing work for disciplinary analysis have had to be done in commercial CAD system or FEA preprocessing software. The entire panel definition and analysis process takes up to several hours. Significant time and cost reductions have been accomplished using the panel ADEE.

The panel ADEE does not seek to automate the entire panel design process, for example, the definition of manufacturable skin layers is not automated by the panel ADEE, but left to a panel manufacturing engineer who will have the required manufacturability knowledge. As a design aided tool, the panel ADEE enhances the design speed by automating the repetitive panel design actions, and panel cost, weight, and the stress state of the panel structural members can be quickly evaluated. This accelerated panel design speed, makes the panel deign more flexible and more capable of reacting to the changes of global fuselage design.

There are some areas for improvements in the current design system. One, there are no iterations between global level and local level. Panel stiffness from the local panel design may be no longer consistent with the global model. The global model should adapt to the stiffness changes from the local level. Then, a new panel design should be defined, and cost and weight should be re-estimated/evaluated, the panel should be re-analyzed structurally, and so forth, until the panel stiffnesses of the global and local models converge.

Two, the panel cost estimation was not validated for this research because cost data is extremely difficult to access, however, the bottom-up parametric cost estimation model still offers an excellent demonstration of the advantages of integration of cost estimations based on geometric parameters provided by a panel MMG.

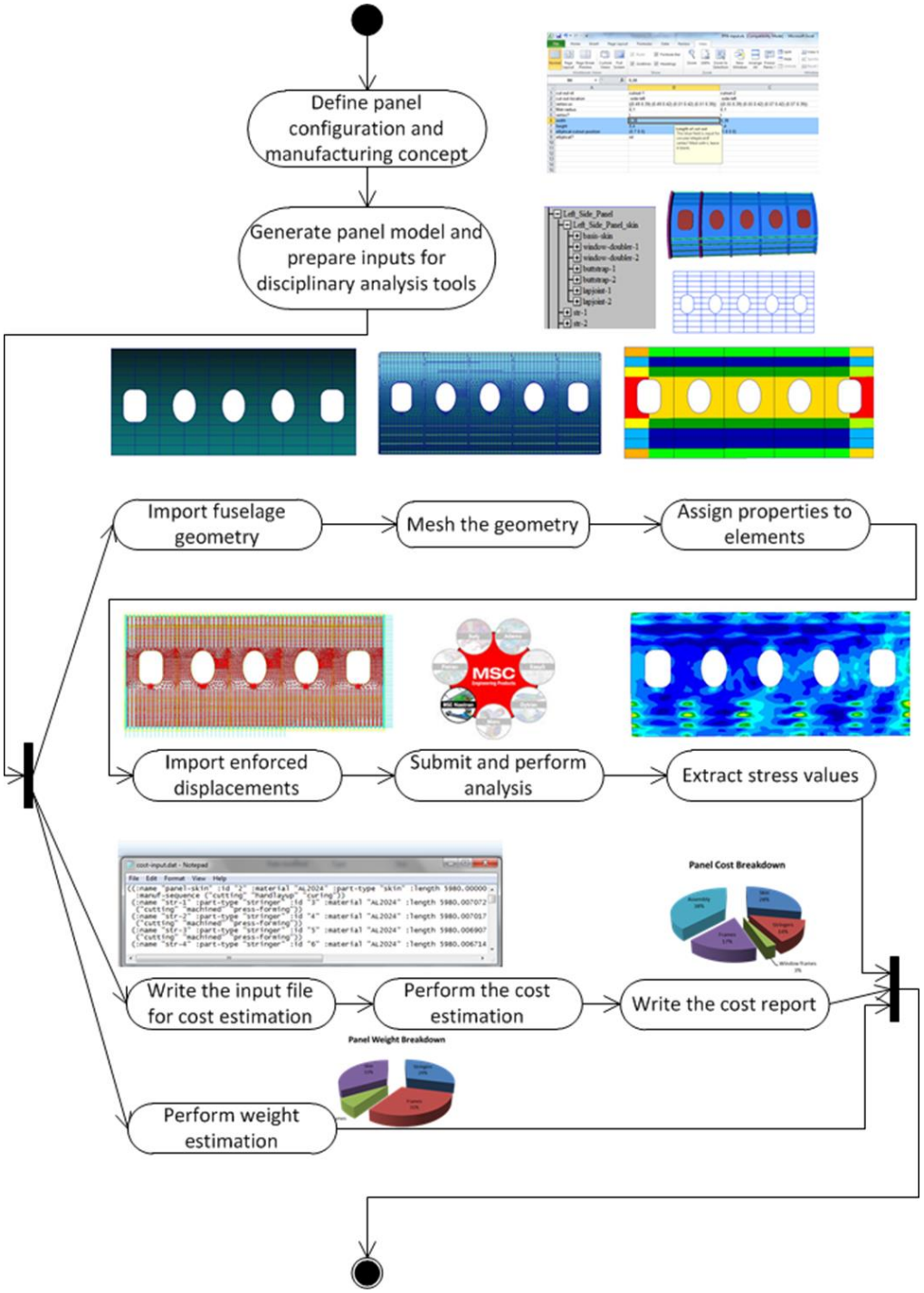


Figure 5.23: The automated structural analysis, cost and weight estimation process

Chapter 6. Implementation and verification of the movable ADEE

The implementation detail of the movable ADEE is described in this chapter. This KBE-supported local subsystem design system permits suppliers to explore large design spaces and search for optimal designs for airframe subsystems in a short lead time. The outline of this chapter is as follows: the overview of the movable ADEE is described in Section 6.1, including the formulation of the cost/weight multi-objective optimization problem. The implementation of the FEA-based weight estimation method, which uses FEA to calculate the internal stresses of the structural elements and an analytical composite plate sizing method to determine their minimum required thicknesses is discussed in Section 6.2. The implementation of the movable MMG is described in Section 6.3, the flexible modeling capability of which is demonstrated by generating different configurations of aircraft movables. The application of the ADEE to an aircraft rudder, from which the Pareto optimal set of structural configuration for minimum cost and minimum weight has been successfully found, is discussed in Section 6.4.

6.1 Overview of the movable ADEE

The movable ADEE was developed on the basis of the requirements discussed in Sub-section 3.4.3, and its schematic overview of the movable ADEE is shown in Figure 6.1. The functionality of each component within the movable ADEE is described below.

The *Initiator* is a user interface module, which allows designers to define a first configuration of an aircraft movable, such as the number of spars, ribs, etc. The configuration is then fed to the optimization loop.

The *movable MMG* is a parametric and generative movable model that can model different types of movable configurations, by combining and scaling a limited amount of high level primitives (HLP), such as spars, ribs, hinges, etc. For the clear definition of a basic movable model, the reader is referred to Sub-section 6.3.1. A set of capability modules (CM) was developed to take care of processing the movable models to extract and assemble, in the right format, the data required for the analysis tools, i.e. FEA-based weight estimation and cost estimation, integrated in the framework.

The *Analysis tools* include the manufacturing cost estimation tool described in Sub-section 5.2.2, and the FEA-based weight estimation method discussed in Section 6.2. This weight estimation tool uses finite element analysis to calculate the internal stresses in the structural elements, and an analytical composite plate sizing method to determine the minimum thickness of the structural elements, under the calculated internal stresses.

The *Optimizer* uses the MMG to search for the best structural topology of the movable, based on the evaluation results from the disciplinary analysis tools. A cost/weight multi-objective optimization was set up, where the design variables were those which determine the topology of the given movable structure, such as the number and the position of the various spars and ribs. The rib and spar numbers are integer variables, whereas the rib and spar positions are continuous variables. Therefore, the optimization problem is a mixed-integer programming problem. The used optimization algorithm is called *ga*, a MATLAB function in the optimization toolbox. The Optimizer searches for the best

layout of the movable structures of an aircraft within a given manufacturing concept, i.e. part production method and assembly methods are specified before running the optimization. The optimization problem for cost and weight study is formulated as follows.
Given: movable outer mould line (OML), manufacturing concept, load cases

Minimize: direct operating cost $DOC = cost(X) + P * weight(X)$ (6.1)

By varying: *design variables X*, i.e. spar and rib numbers, spar and rib position
Subject to:

1) no material failure in any structural member
2) no buckling failure in any structural member under 100% limit load

The coefficient P included in the objective function is the weight penalty parameter, which is defined as the ***price paid by OEM per kilogram weight saving*** (in €/kg). The weight penalty approach is introduced in Kelly et al. (2002) and used in Curran et al. (2006) to estimate DOC, on the basis of manufacturing and material cost and total structure weight. The quantification of P is not trivial; in this work, a typical value of 500$/kg was initially assumed for the composite structures, as suggested in Kelly et al. (2002), and then this value was from 0 to 10,000 varied during the multi-objective optimization.

The *Communication framework* was implemented in MATLAB to coordinate the data exchange between any two components mentioned above. Note: for a complete overview of the dependencies and data exchanges between the various components, the reader can refer to the N2 chart provided in Figure 6.2.

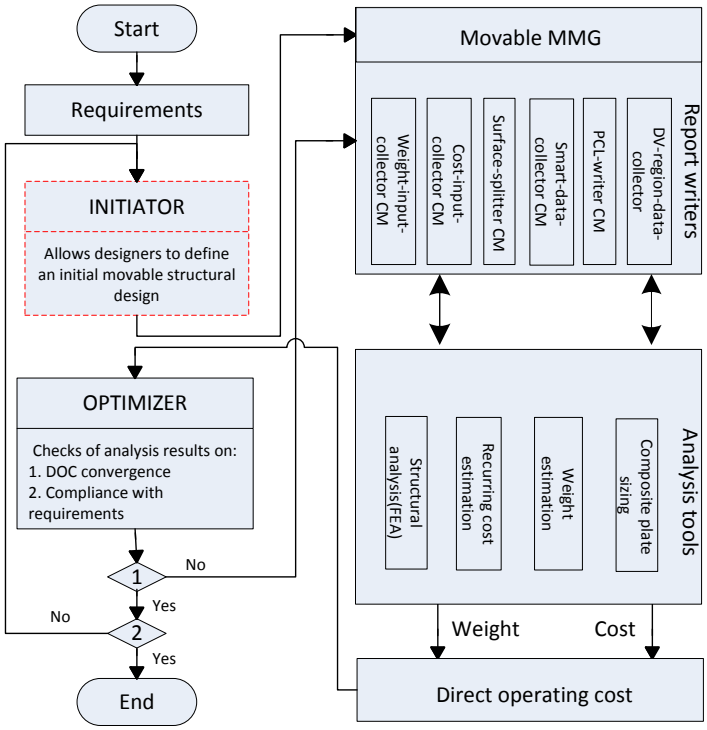


Figure 6.1: Schematic overview of the movable ADEE, the component in the dashed block is not implemented for this ADEE

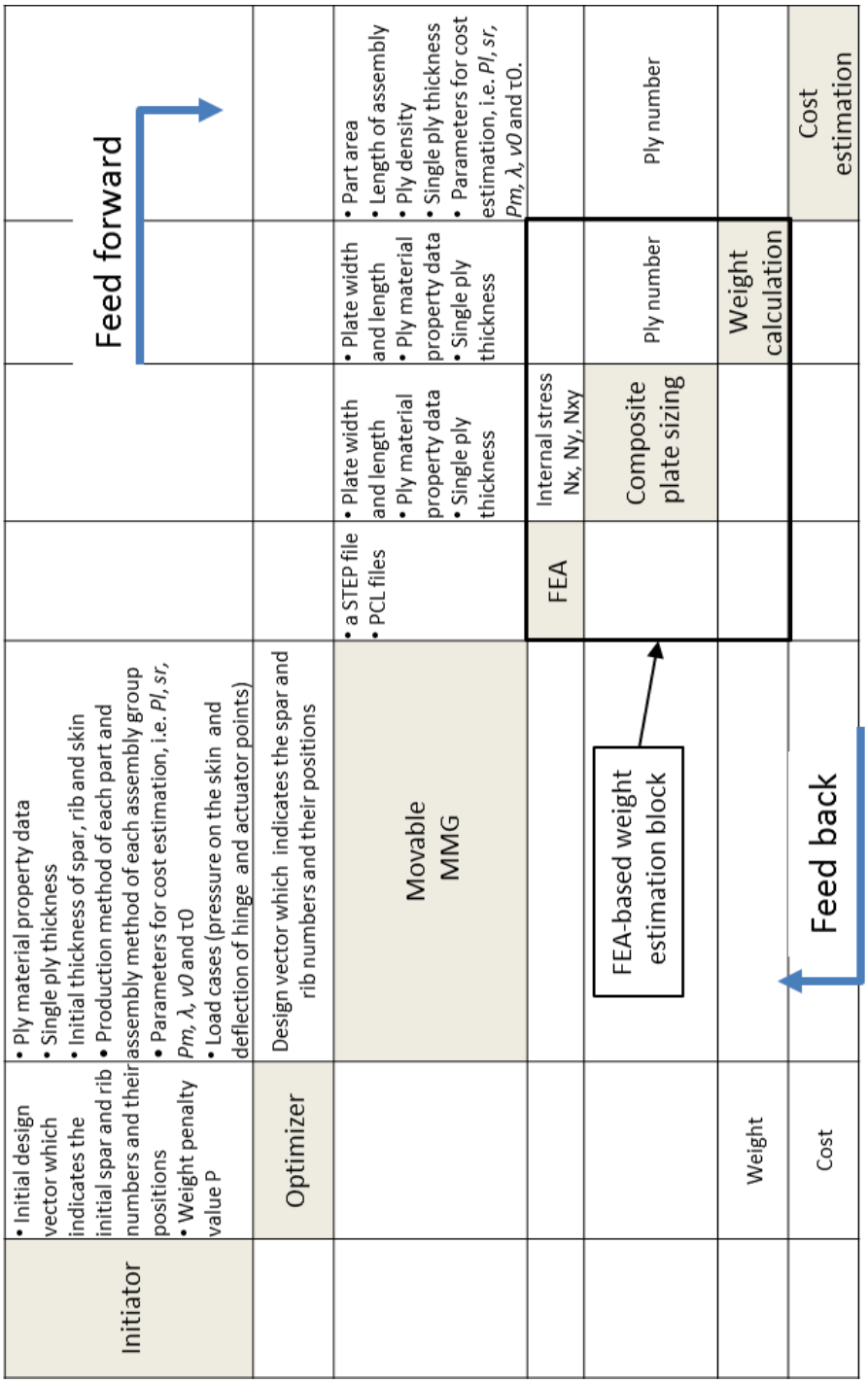


Figure 6.2: Movable ADEE N2 chart

6.2 FEA-based weight estimation method

The total weight of movable structures is decomposed into the weight contribution from the primary and secondary structures, such as sealing, painting, fasteners and hinge brackets. Only the primary structural weight is considered in the movable ADEE, and this is calculated by aggregating the weight of each component which is determined by the material volume and density of the component. Therefore, the component's thicknesses have to be determined to calculate the material volume. The weight estimation method implemented in the movable ADEE uses FEA to calculate the internal stresses of the structural elements and an analytical composite plate sizing method to determine their minimum required thicknesses.

In contrast to the sizing method implemented in the fuselage ADEE, there is no iteration in the sizing method used by the movable ADEE, however, initial sizing studies done for the research reported here demonstrated that the differences between the iterative and non-iterative sizing methods are not significantly big, hence the non-iterative sizing methods were not considered adequate for preliminary sizing. In addition, the non-iterative method has a better performance than the iterative one with respect to (w.r.t.) the computational time.

A sizing method for composite laminate plates was developed to determine the thickness values of the plates, skin panels, spar webs and ribs. The failure modes considered by the composite plate sizing method are described below.

Material strength failure

According to classical laminate theory, the composite applied strain $\bar{\varepsilon}$ and curvature κ can be calculated from forces N and moments M as follows:

$$\begin{pmatrix} \bar{\varepsilon} \\ \kappa \end{pmatrix} = \begin{bmatrix} A & B \\ B & D \end{bmatrix}^{-1} \begin{pmatrix} N \\ M \end{pmatrix} \tag{6.2}$$

where A is the extensional stiffness, B is the coupled stiffness and D is the bending stiffness of the laminate. The symmetrical stacking sequence were chosen such that the coupled stiffness B was 0. The loads on the structure, shear forces, bending moments, and torsional moments, were translated into in-plane loads of tension, compression, shear, or a combination of these. Then, Eq. 6.1 could be reduced to:

$$\begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix}^{-1} \begin{pmatrix} N_x \\ N_y \\ N_{xy} \end{pmatrix} \tag{6.3}$$

where N_x, N_y, N_{xy} are in-plane running loads extracted from the elements of the FE-model, and A_{11}-A_{66} are the components of the extensional stiffness A. The principal applied strains can be calculated as follows:

$$\varepsilon_1 = \frac{\varepsilon_x + \varepsilon_y}{2} + \frac{\sqrt{\left(\varepsilon_x - \varepsilon_y\right)^2 + {\gamma_{xy}}^2}}{2} \tag{6.4}$$

$$\varepsilon_2 = \frac{\varepsilon_x + \varepsilon_y}{2} - \frac{\sqrt{(\varepsilon_x - \varepsilon_y)^2 + {\gamma_{xy}}^2}}{2} \tag{6.5}$$

The material allowable strains were calculated by multiplying room temperature ambient (RTA) strength with "knockdown" factors considering the "worst of all situations", such as the low end of the strength distribution, the worst type of damage, and the worst operating environment. If the material RTA strength and "knockdown" factors are not available, the suggested allowable strains given in Table 6.1 can be assumed for the conceptual design (Elham et al, 2013).

Table 6.1: Suggested design allowable for composites

Tension (μs)	Compression(μs)	Shear(μs)
5000	4500	6000

Plate buckling under compression and shear

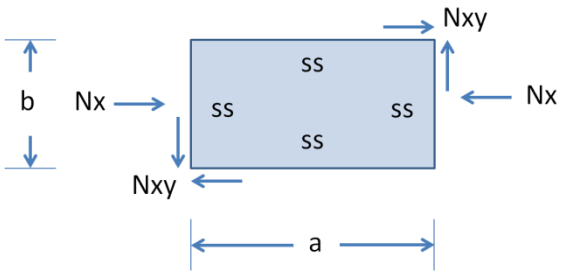


Figure 6.3: A rectangular composite laminate plate under shear and compression

Each plate was checked for buckling under shear and compression using the following formula (Kollár et al., 2003)

$$\frac{N_x}{N_{xcrit}} + \left(\frac{N_{xy}}{N_{xycrit}}\right)^2 \leq 1 \tag{6.6}$$

where

$$N_{xcrit} = \frac{\pi^2 \left[D_{11}m^4 + 2(D_{12} + D_{66})m^2(\frac{a}{b})^2 + D_{22}(\frac{a}{b})^4\right]}{a^2 m^2} \tag{6.7}$$

$$N_{xycrit} = \pm\frac{4}{a^2}\sqrt[4]{{D_{11}D_{22}}^3}(15.07 + 7.08K) \;\; when\; K \leq 1 \tag{6.8}$$

$$N_{xycrit} = \pm\frac{4}{a^2}\sqrt[4]{D_{22}(D_{12} + 2D_{66})}\left(18.59 + \frac{3.56}{K}\right) \;\; when\; K > 1 \tag{6.9}$$

where K is defined as follows

$$K = \frac{D_{12} + 2D_{66}}{\sqrt{D_{11}D_{22}}} \tag{6.10}$$

The boundary conditions of the plates within the open leading edges (see Figure 6.4) were assumed to be one free edge and three simply supported edges. Then the critical compression buckling loads are given (Kassapoglou, 2010) by:

$$N_{xcrit} = \frac{\pi^2}{b^2}\sqrt{D_{11}D_{22}}\left(\frac{12}{\pi^2}\frac{D_{66}}{\sqrt{D_{11}D_{22}}} + \frac{b}{a}\left(\frac{D_{22}}{D_{11}}\right)^{1/4}\right) \tag{6.11}$$

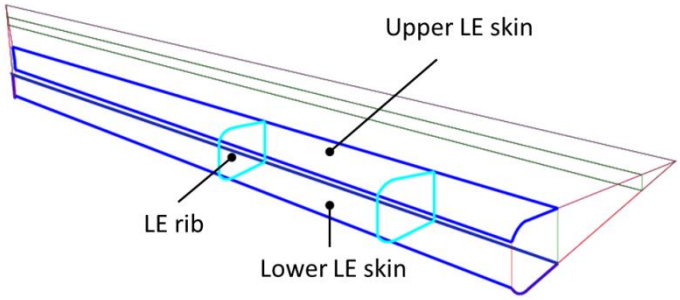


Figure 6.4: Open leading edge skins. The upper LE skin and the lower skin are not connected

All the laminates were assumed to be midplane symmetrical, with a predefined $[+45/-45/90/0]_s$ stacking sequence. This stacking sequence is commonly used in design practice to limit the large amount of variables possible when dealing with laminate stacking. The thickness of 0°, ±45° and 90° layers are design variables. Note: the thickness of +45° and -45° ply are equal due to symmetry. The lower limit of the thickness variables were chosen such that there was at least one ply in each direction. The thickness of a single ply was set as 0.138 mm. All three thickness variables were continuous to improve the convergence of laminate sizing process. A possible mismatch with the prepreg ply thickness, 0.138mm, was accepted.

Table 6.2: Thickness variables and their lower and upper limit

Variables	Lower limit (mm)	Upper limit (mm)	Description
x1	0.138	10	0° skin thickness
x2	0.138	10	90° skin thickness
x3	0.138	10	45° or -45° skin thickness

The optimization determined the dimensions of all the composite plates in the way the minimum composite plate weight was achieved. The design variables for sizing composite plates are listed in Table 6.2. The optimization problem is formulated as follows.

Given: plate dimensions a, b, boundary conditions, running loads Nx, Ny, Nxy from FEA

Minimize: $W_{plate} = \rho*a*b*2*(x1 + x2 + 2*x3)$

By varying: *x1*, *x2*, *x3*
Subject to:

1) the principal applied strains are less than tension/compression allowable strains
2) the applied shear strain is less than allowable shear strain
3) no buckling failure under 100% limit load

6.3 Implementation of the movable MMG

The requirements on the capability of the movable MMG were defined in Sub-section 3.4.3 and are summarized below.

- The movable MMG should be able to generate the geometric entities representing the structural members that appear in the majority of movable components
- The movable MMG should be able to generate the OML surface in case the OML surface is not provided by the OEM
- The movable MMG should be able to generate internal structural layouts of different topology, e.g. multi-spar, multi-rib, etc.
- The movable MMG should be able to prepare the input file for cost estimation
- The movable MMG should be able to automate the sizing workflow, see Table 6.3 for more detailed requirements

Table 6.3: Requirements for the MMG to implement the FEA-based weight estimation

No.	Requirements	Description
1	Generation of meshable surfaces	for automatic FE model generation
2	Automatic assignment of properties	for automatic FE model generation
3	Automatic meshing and elements creation	for automatic FE model generation
4	Loads and boundaries condition application, submission analysis job	for automatic FE model generation
5	Generation of a stress report	providing running loads for composite laminate sizing
6	Provide the material density, width, length and area of the design variable region	providing plate properties for composite laminate sizing

6.3.1 Movable HLPs

The *movable-trunk* HLP models most of the structural members, e.g. spars and ribs, which can be reused for the manufacturing view[3], which will be discussed later. The *movable-trunk* HLP concept is similar to the *wing-trunk* HLP which was first proposed and implemented by La Rocca et al. (2003) to generate the geometric entities of the aerodynamic surface and internal structures of a lifting surface. An example movable that is composed of several instance of *movable-trunk* is given in Figure 6.5. The *hinge* HLP models the hinges and actuators which connect the structural members, skin/ribs/spars, to the lifting surfaces. The UML class diagram of the movable MMG is given in Figure 6.6.

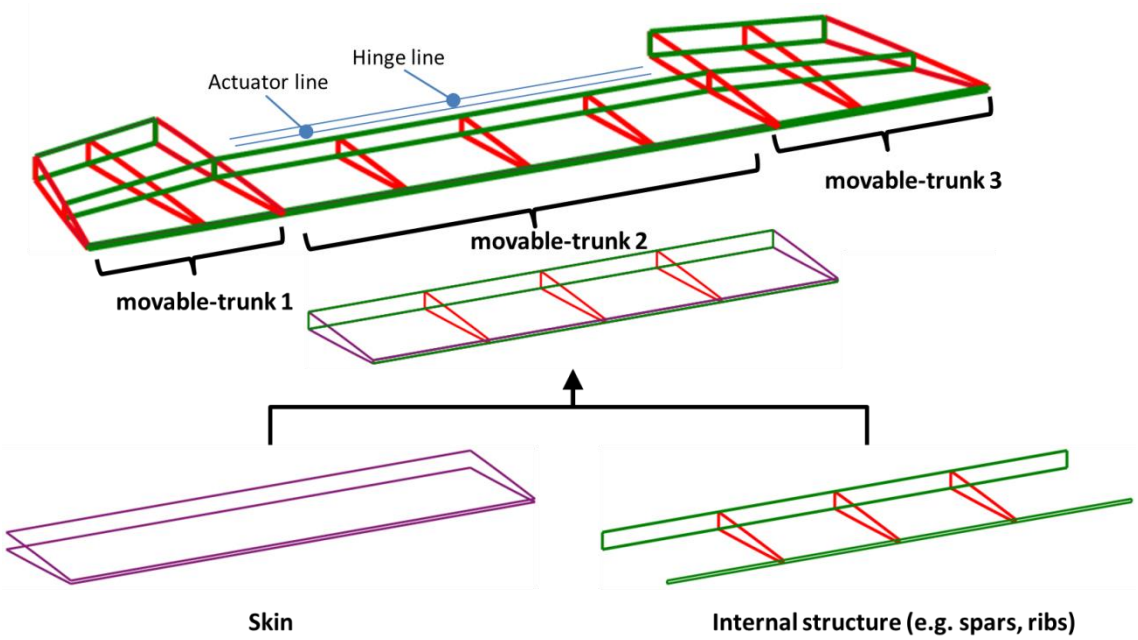


Figure 6.5: Example of complex elevator model composed by three instances of *movable-trunk*. Each trunk instance is composed of the skin and internal structures

The modeling process of a movable structure is shown in Figure 6.7. First, the OML is generated internally or provided as input to the movable ADEE. The OML is generated inside the movable ADEE by reusing the *wing-element-clean* class in DARwing (van den Berg, 2009), which is a KBE application to model the lifting surfaces. Several groups of input parameters are required. Note: the inputs for the movable MMG are described in Appendix I. One group of input parameters is used to define the LE and TE lines. Another group is needed to specifying the types of root and tip airfoils. The third group of input parameters is required to specify how to position these airfoils. When the OML surface is provided by the OEM, six geometrical elements are required by the *wing-element-clean* class: hinge and actuator lines, LE and TE lines, and root and tip airfoils. All of these elements are provided to the movable ADEE in the form of IGES (van den Berg, 2009). Then the MMG imports

[3] As shown in Figure 2.13, domain experts from different disciplines have different views on the same product. The *manufacturing-view* is a collection of classes from which the inputs for cost estimation are derived, whereas the inputs for weight estimation are extracted from the *structural-view*.

these IGES files and re-parameterizes these curves in such a way that the *wing-element-clean* can internally model the OML.

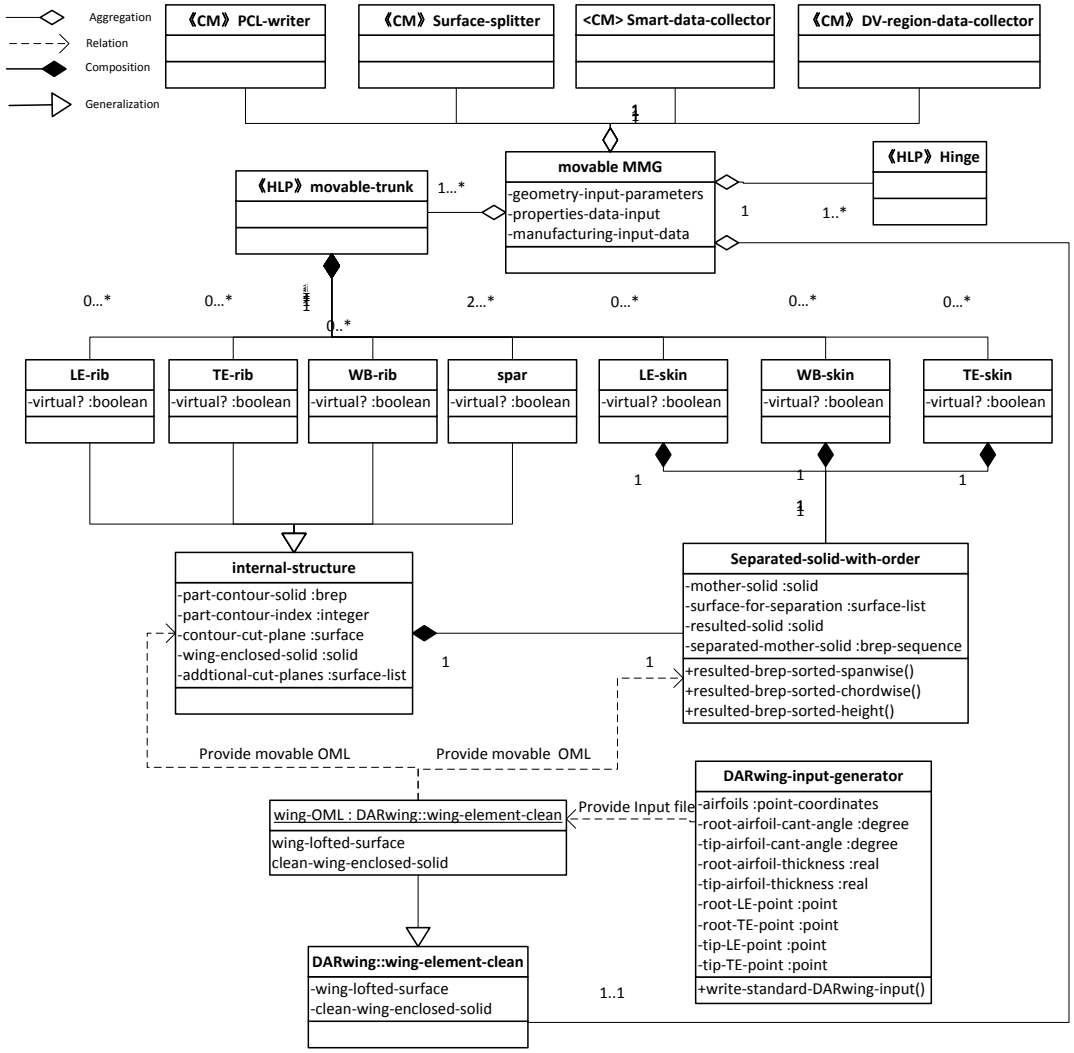


Figure 6.6: UML class diagram used to implement classes relevant to the structural view

After the OML is generated internally or provided outside the movable ADEE, the division of the OML into several trunks has to be done. The division is done by splitting the OML using *trunk-dividing-ribs*. The definition of these ribs is similar to the definition of a rib in a wing trunk, which will be described below. After the OML of each wing trunk is obtained, the required inputs for generating internal trunk structures have to be specified trunk by trunk. All the surfaces of the internal structures are generated by intersecting a surface with the movable OML, this can guarantees that the contours of internal structures lie in the OML surface. The normal vector of each cut plane is either user-defined or automatically calculated when the definition of a structural member refers to other internal structures. An *internal-structure* class was implemented to obtain the geometrical

representation of the internal structures. An instantiation of *separated-solid-with-order*[4] class is included in each instantiation of *internal-structure* to split the contour into meshable surfaces. See Section 6.3.2 for more details about segmentation.

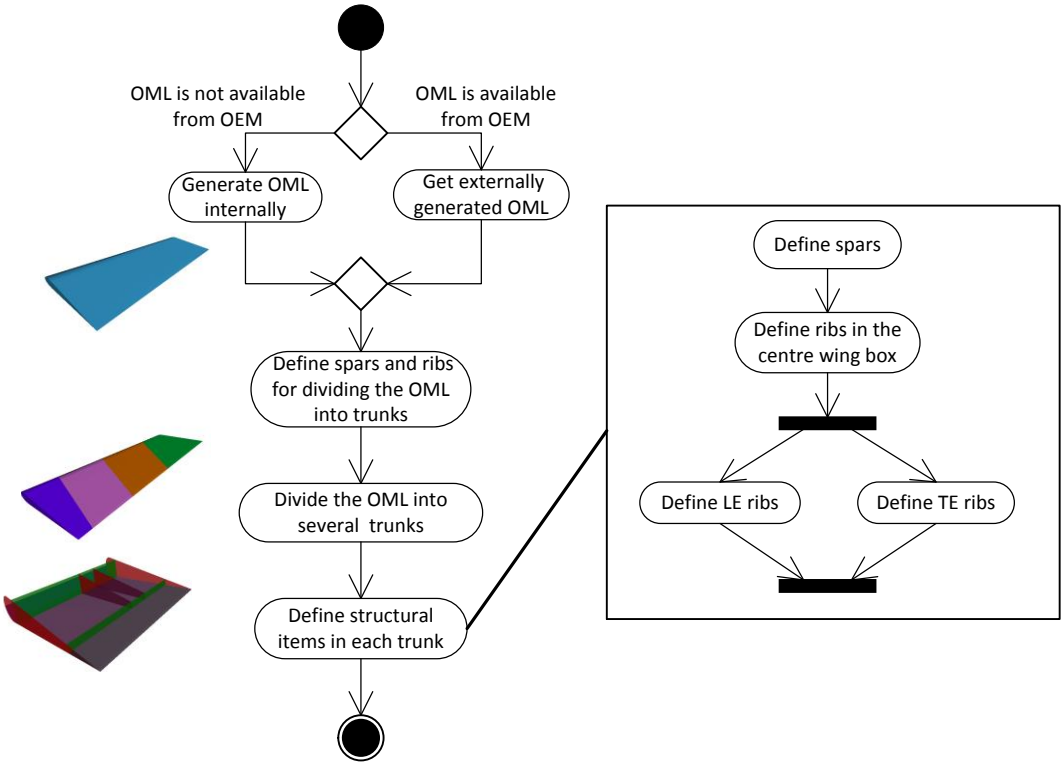


Figure 6.7: Activity diagram of defining the movable structures of an aircraft

The definition and modeling process of the entities in a movable trunk is described below.

Spar

A spar is generated by the following sequence of activities, see Figure 6.8 for illustrations of these activities.

- Two points are located at the root chord and the tip chord respectively. The line called the spar line is constructed from the point at the root chord to the point at the tip chord.
- Going through the spar line, a plane is constructed with V1 as its normal vector, which is perpendicular to the YoZ plane[5]. The plane intersects with the watertight solid derived from the movable OML.
- A start and end rib can be specified to define partial spars, i.e. spars that do not extend from root to tip.

[4] Several segments can result from performing a Boolean operation to the surfaces representing structural members. The *separated-solid-with-order* sorts the segment according to their spanwise or chordwise position by comparing their vertex coordinates.

[5] The datum coordinate system in Figure 6.8 for defining a spar is a user defined coordinate system. X axis is along spanwise while Y as axis is along chordwise. The origin of the coordinate system locates at the leading edge point of the root airfoil.

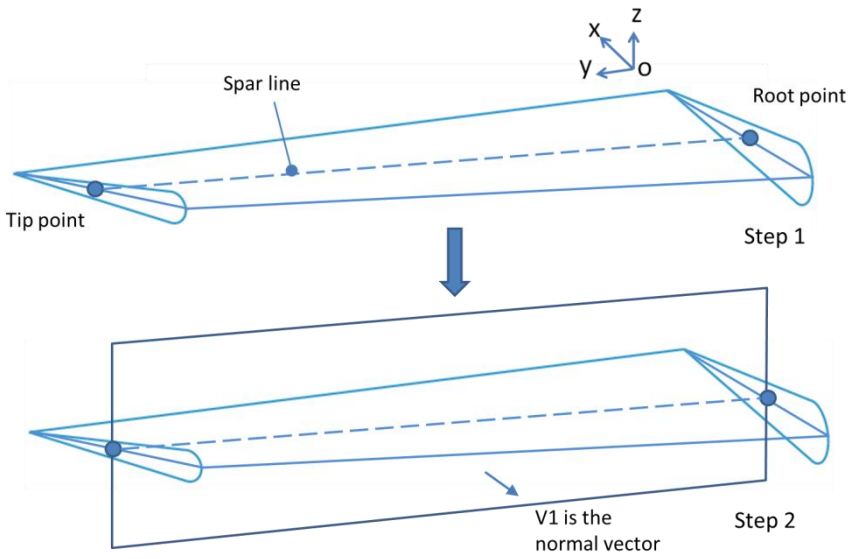


Figure 6.8: Generation process of the spar

The input parameters required to generate a spar are listed in Table 6.4. Virtual spars can be instantiated for the purposes listed below:

- to provide the datum to other structural items
- to create meshable surfaces during segmentation process
- to refine the mesh

Table 6.4: Input parameters for spar definition

Parameter	**Example**	**Description**
:root-chord-percentage	0.25	The parameter range is [0 1].
:tip-chord-percentage	0.25	The parameter range is [0 1].
:start-rib-index	1	A rib identification number. This parameter specifies the rib from which the partial spar starts.
:end-rib-index	3	A rib identification number. This parameter specifies the rib at which the partial spar ends.
:virtual-spar?	t	Default value is nil. t: virtual spar; nil: real spar. A virtual spar is used to generate meshable surfaces or provide reference for defining other structures.

Rib

There are three different types of ribs that can be modeled by the *movable-trunk* HLP, namely the wing box (WB) rib, the leading edge (LE) rib and the trailing edge (TE) rib. The WB ribs are defined first and they are used as references for the LE ribs and TE ribs.

A WB rib is generated using the following three steps, as illustrated in Figure 6.8.

- specify a point which the rib cut plane[6] goes through. A reference spar is first selected via the parameter *positioning-ref-spar-index*, then the point on the reference spar line is specified by the parameter *rib-cut-plane-center*, See Figure 6.9 (a)
- specify the orientation of the rib cut plane, first, select the reference spar via the parameter *:orientation-ref-spar-index*, then, the rib orientation can be specified as perpendicular to the spar line of the reference spar or specified with an angle between the rib cut plane and the X axis
- a contour is generated by intersecting the rib cut plane with the movable OML, then the part between the front and rear spar is kept as a contour of the WB rib

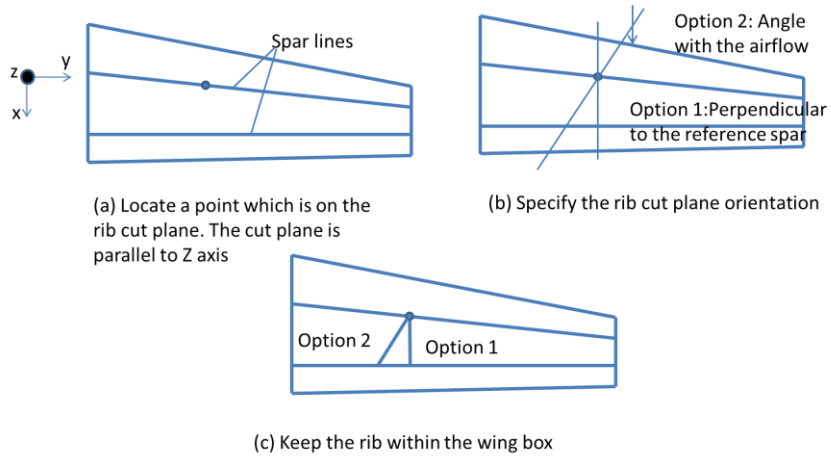


Figure 6.9: Generation process of the WB rib

The input parameters for a WB rib are listed in Table 6.5.

Table 6.5: Input parameters for a WB rib.

Parameter	Example	Description
:orientation-ref-spar-index	1	A spar identification number. Chosen from [0 (sparNum - 1)].
:perpendicular-to-ref-spar?	t	t: choose option 1 for rib definition; nil: choose option 2 for rib definition.
:positioning-ref-spar-index	1	A spar identification number. The spar line of Spar 1 is chosen for positioning the point which is on the rib cut plane. Chosen from [0 (sparNum - 1)].
:rib-cut-plane-center	0.25	Chosen from [0 1]. 0: a point on this rib locates at the start of a spar

[6] The rib cut plane is a plane used to cut the watertight solid derived from the movable OML to get the rib contour surface. Similarly, the surfaces used to obtain spar contour surfaces are called “the spar cut planes” in the reminder of this thesis.

		line. 1: a point on this rib locates the end of the spar line.
:angle-with-streamwise	10	Degree. Chosen from [-90° 90°].
: virtual-rib?	nil	nil: real rib; t: virtual rib.
:attached-hinge?	t	nil: not attached to a hinge; t: attached to a hinge.
:attached-actuator?	t	nil: not attached to an actuator; t: attached to an actuator.
Notes: - The virtual rib is used to provide the datum for LE and TE ribs or to improve the mesh quality of the WB skin. - When *: perpendicular-to-ref-spar?* = t, the value of *:angle-with-streamwise* is not relevant.		

A LE / TE rib is defined after the definition of the WB-ribs is complete. A LE/TE rib must attach to a WB rib. The orientation of a LE / TE rib cut plane is defined by rotating the cut plane of the attached WB rib by a specified angle. Example inputs of defining a spar, a WB rib, a LE rib and a TE rib are illustrated in Figure 6.10.

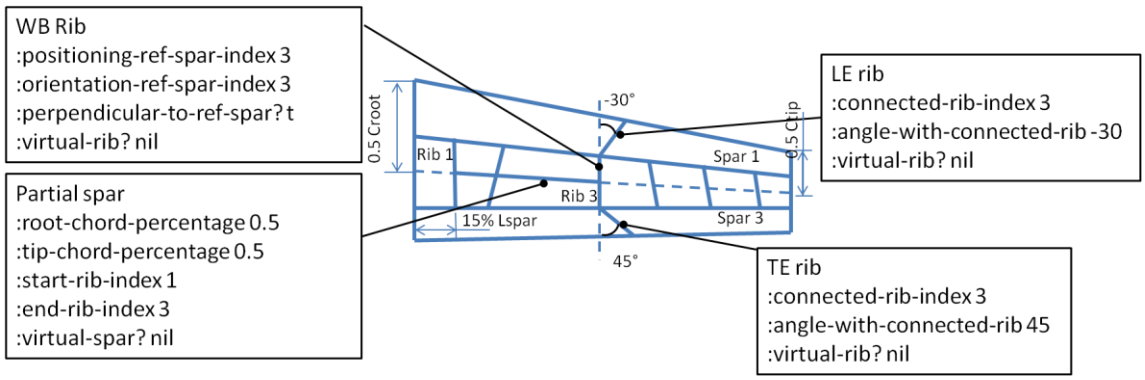


Figure 6.10: Example of spar, WB, LE and TE ribs definition

Hinge and actuator

The generation process for a hinge is shown in Figure 6.11. In order to define a hinge, a hinge line is constructed first as the positioning reference. The start point of the hinge line is positioned by specifying the parameters, *offset-in-x-root* and *offset-in-z-root*, which are offsets from the LE point of the root airfoil along x axis and z axis respectively. The end point of the hinge line is positioned in the same way as the start point, then, the hinge line is intersected with the cut plane of the rib which the hinge attaches to, and a hinge point is obtained as the geometric representation of the hinge. Finally, two connection lines are generated to attach the hinge point to the two vertices of the most forward edge of the hinge rib. The same modeling approach is implemented to define actuators.

A unique number is given to each hinge and actuator such that for each load case the user can directly define the translation and rotation of the hinge/actuator points via the input parameter *node-*

displacement, shown in Appendix H, for the movable MMG. When building the movable FE model, the coordinates of the hinge and actuator points are obtained from the product model. Nodes will be created accordingly at the hinge and actuator positions with the specified identification number. Rigid body elements (RBE2) are used to connect the hinge/actuator nodes with the two nodes that represent the two vertices of the most forward edge of the hinge/actuator attaching rib.

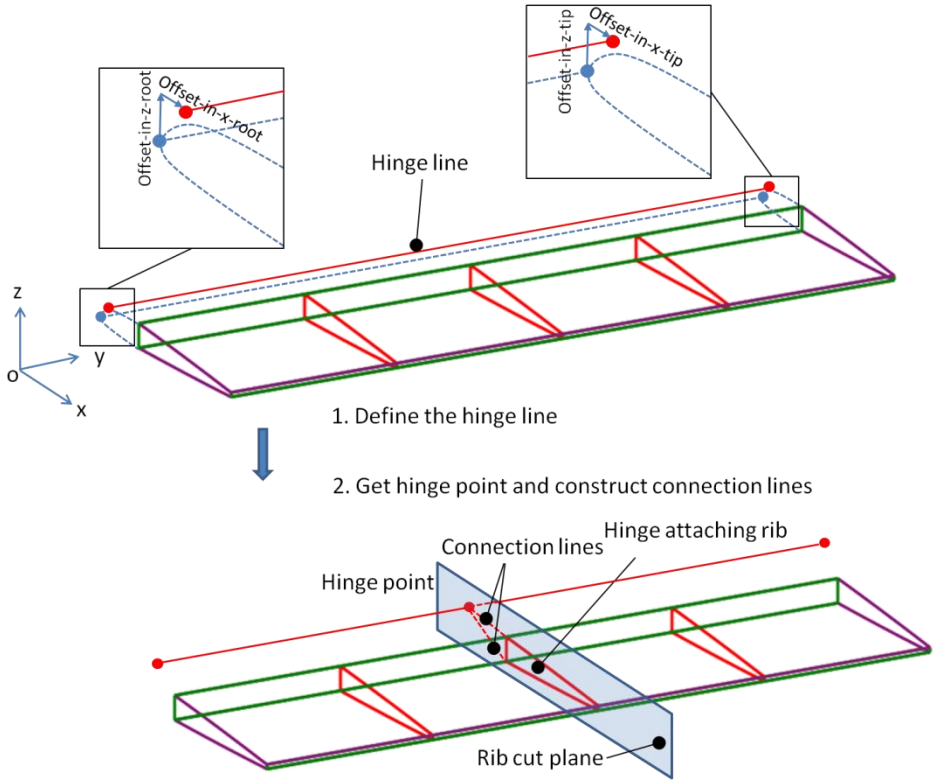


Figure 6.11: The hinge generation process

Figure 6.12 shows example movables layouts generated by the movable MMG.

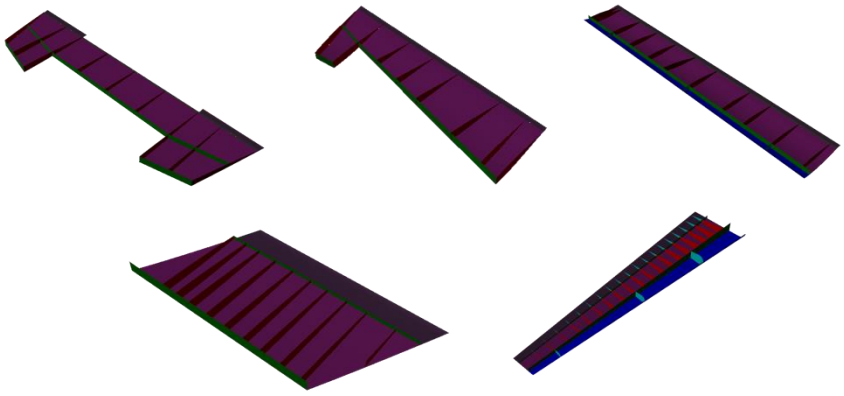


Figure 6.12: Example of movable models

Modelling limitations

Although the MMG has sufficient flexibility to model many movable structures, its current implementation has some limitations:

- the spar definition only supports the "point-to-point approach". A spar cannot be generated with the "point-to-angle" approach. These two definition approaches are shown in Figure 6.13.
- curved spars/ribs cannot be generated, see Figure 6.14.
- spars and ribs are not allowed to cross over in one movable trunk, see Figure 6.14.
- minimum two spars are required for a movable trunk. These two spars can be modelled as two virtual spars. One spar is modelled as real and one is virtual for a movable using only one spar.

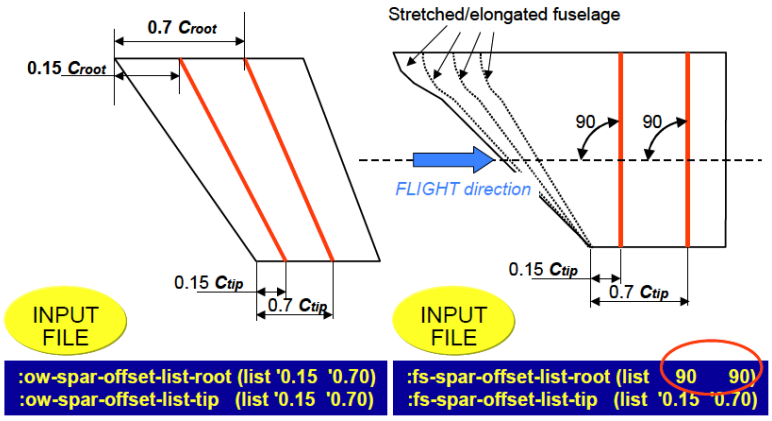


Figure 6.13: Two spar positioning approaches: point-to-point, left, and point-to-angle, right (La Rocca, 2010)

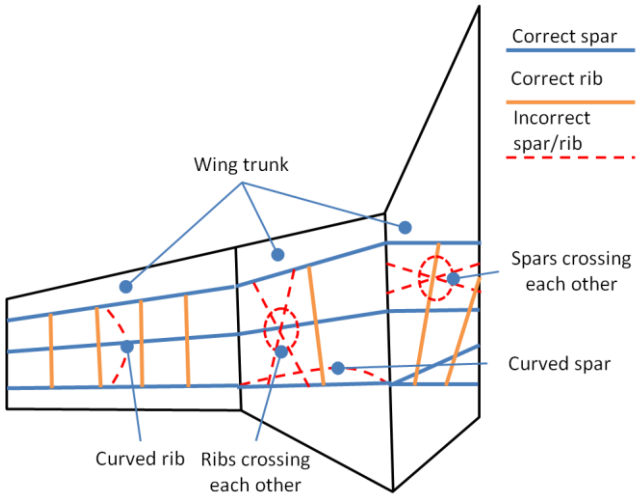


Figure 6.14: Valid and wrong definitions for spars and ribs

Implementation of manufacturing view

After the geometric entities, i.e. movable trunk spars and ribs, for each movable trunk have been generated, the *Manufacturable-part* class is used to model manufacturable parts by reordering the geometric entities within movable trunks. More parameters such as production method and material are specified as non-geometrical attributes to instantiate *Manufacturable-part*. In Figure 6.15, the

contour of a spar is shown in the dashed area, which consists of three geometric entities: the second spar in the movable-trunk 1, the first spar in the movable-trunk 2 and the first spar in the movable-trunk 3.

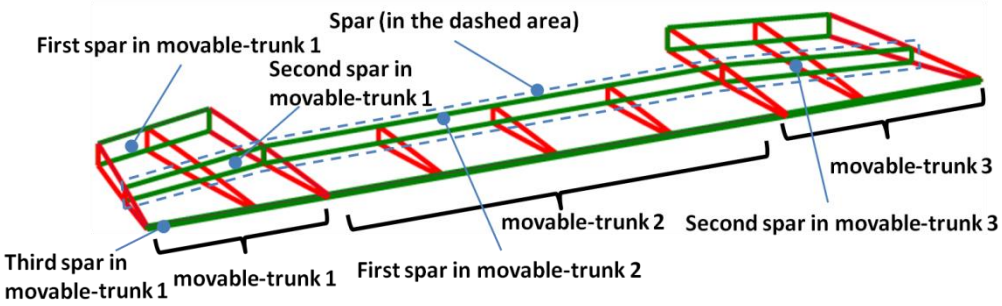


Figure 6.15: A spar assembled by three entities of different movable trunks

The instantiations of *Manufacturable-part* and *Assembly* together with the *cost-input-collector* CM form a manufacturing view on which the inputs for cost estimation are based. The UML class diagram of the manufacturing view is given in Figure 6.16.

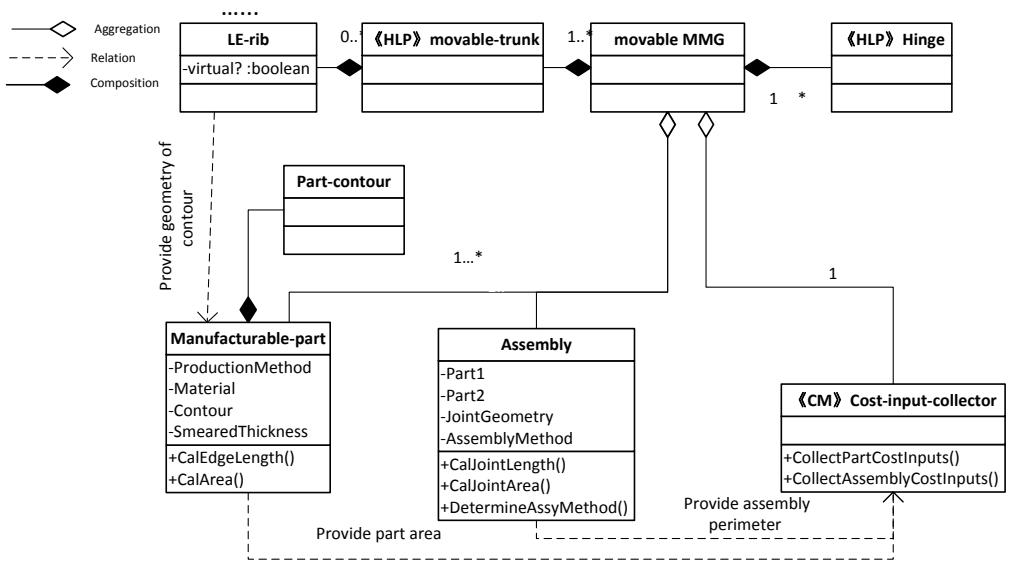


Figure 6.16: UML class diagram used to implement classes relevant to the manufacturing view

The *Assembly* class is used to model the assembly joints that are geometrically represented by lines. The assembly method is added to the *Assembly* class as a non-geometrical attribute of the *Assembly* class, and usually is defined as an input. In practice, the assembly method for movable structures is determined according to the material types of the parts that will be assembled together. The one-by-one definition needs a lot of human input which prohibits the use of automation in the optimization process. To solve this problem, a database has been built to determine the assembly methods according to the part material types (van der Laan, 2008), an example of which is given in Figure 6.17. The database used for this process is maintained and updated by manufacturing engineers to account for new assembly techniques.

The assumption was made during this research presented here that the skin, ribs and spars are manufactured separately, and then assembled to form an aircraft movable. The *Assembly* class, therefore, does not support highly integrated parts such as all integrated wing box using, for example, resin transfer moulding. Under this assumption assembly joints can be divided into three groups, which are listed in Table 6.6. Taking the *Rib-skin-assy* as an example, the edges of the rib contour surfaces that are on the skin contours are extracted to represent the assembly joints between skin and the ribs.

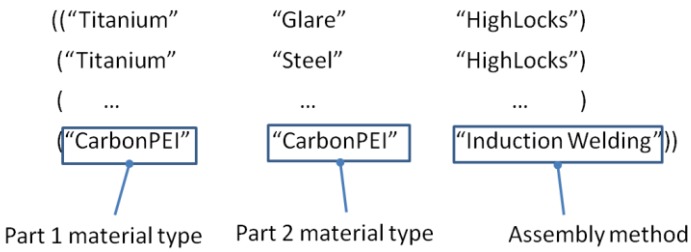


Figure 6.17: A TXT file as a manufacturing database to determine the assembly methods

Table 6.6: Assembly groups between the manufacturable parts

Manufacturable part 1	Manufacturable part 2	Assembly groups
Rib	Skin	Rib-skin-assy
Rib	Spar	Rib-spar-assy
Spar	Skin	Spar-skin-assy

6.3.2 CMs for weight estimation

Three CMs have been developed to enable the automatic FE model generation, namely the *surface-splitter*, the *smart-data-collector* and the *PCL-writer*.

The *surface-splitter* CM splits the structural members into meshable surfaces. The segmentation process automatically performed by this CM is as follows:

- ribs are split by the spar cut planes, including the cut planes of the virtual spars
- spars are split by the rib cut planes, including the cut planes of the virtual ribs
- the LE/WB/TE skin is split by the cut planes of spars and ribs, real and virtual

It should be noted that spars and ribs, the cut planes of which are used for splitting, are the entities in the movable trunks, not manufacturable parts. All the meshable surfaces obtained by the segmentation process are collected in a list, and each one is assigned a unique number for the purpose of traceability. Finally the *surface-splitter* generates a STEP file containing all the segments ready for meshing, which will be read by PATRAN.

The *smart-data-collector* CM, discussed in Chapter 5, is used to extract from the product model the structural properties data that are required by PATRAN for property assignment. The boundary conditions data is directly defined by specifying the deflections at hinges and actuators. These three

data sets are used by the *PCL-writer* to generate PATRAN session files, which automatically automate the generation of the complete FE model.

The *PCL-writer* CM also writes a session file to extract the stress values of elements within each segment for all the considered load cases, described in the application case in the next section. These stress values are multiplied with the plate thickness to get the running loads for sizing for local components. An execution example of the automated FE model generation is shown in Figure 6.18.

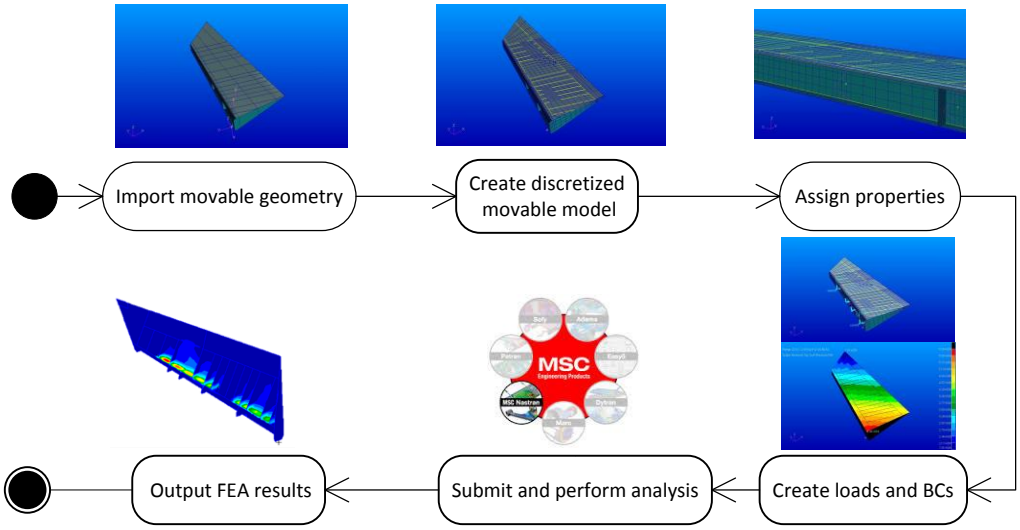


Figure 6.18: An example of the automated FE model generation

Rather than sizing individual segments, the composite plate sizing tool sizes the the design variable (DV) regions. Finite elements within one DV are assigned the same property. The DV regions are obtained by splitting all the surfaces of the movable model with the cut planes of the **real** structural members. Note that this splitting is different from that performed by the *surface-splitter* to obtain the meshable segments, where cut planes of both **real** and **virtual** structural members are used. The *DV-region-data-collector* CM collects the inputs related to the design variable (DV) regions, e.g. length, width and running loads, for composite plate sizing, where elements within one DV are assigned with the same property.

As seen in Figure 6.19, a DV might contain several segments, after the segmentation process, all the surfaces belonging to the various DV regions are collected into the *DV-region-list*. The *DV-region-data-collector* CM iterates each DV region in the list to collect the IDs of segments within the region, from which the running loads are retrieved to size the DV region using the composite plate sizing method described in Section 6.2. Finally, the *DV-region-data-collector* outputs the dimensions of each DV, the density of material and the running loads.

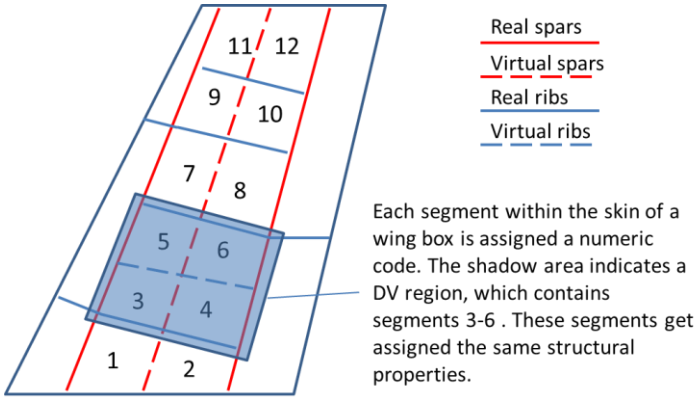


Figure 6.19: A design variable area containing several segments

In the sizing process, different boundary conditions are assumed for DV regions of different movable structural members. The boundary conditions of spars, wing box ribs and wing box skins are assumed to be simply supported edges, whereas those of LE/TE ribs and skins are assumed to be one free edge and three simply supported edges.

The *Weight-input-collector* CM collects the values of area, sized thickness and material density into a text file which are used for the final weight calculation for each instance of the *Manufacturable*-part.

6.3.3 CMs for cost estimation

The *Cost-input-generator* collects all the information into the format required by the cost analysis module, the inputs exacted from the MMG are illustrated in Figure 6.20, and how these data are collected is shown in Figure 6.21.

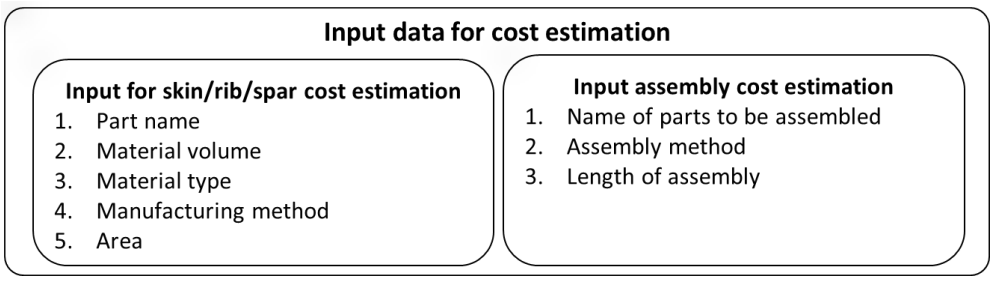


Figure 6.20: Input data required for movable cost estimation

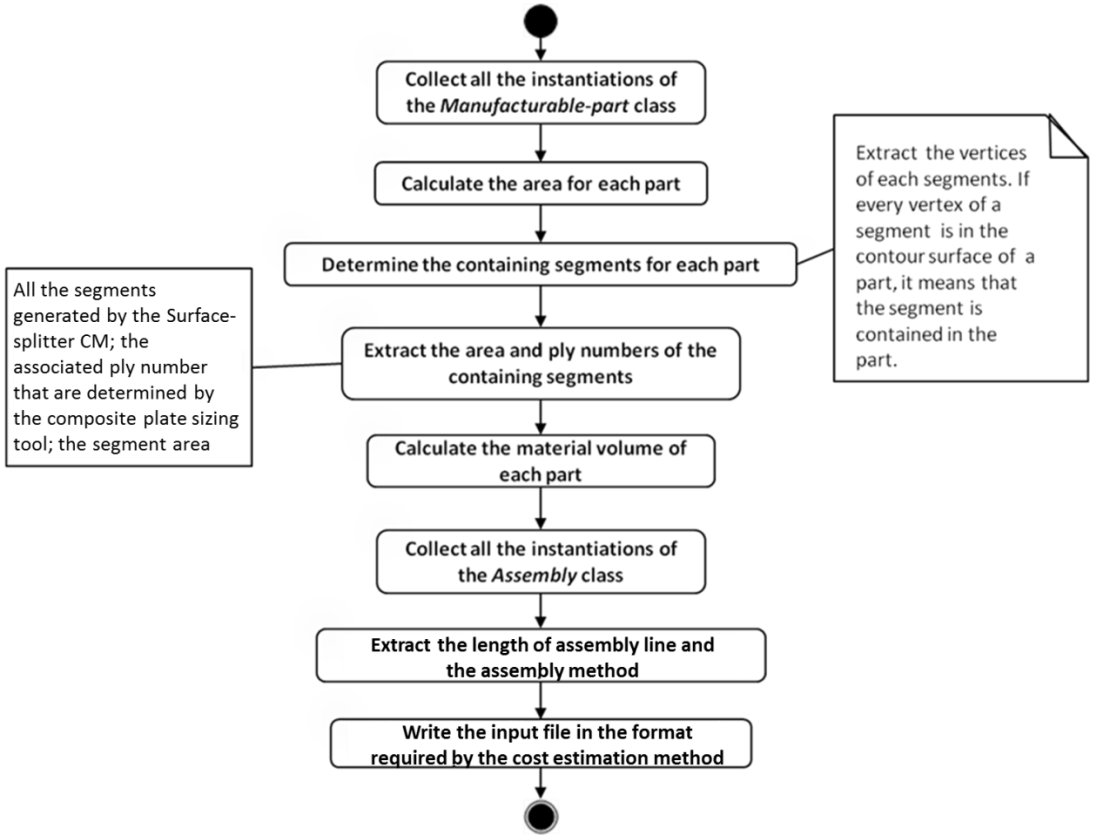


Figure 6.21: Activity diagram of writing the input file for cost estimating

6.4 Application case and discussion

The rudder of a business jet is an airframe subsystem, which is designed by a component supplier. In the RFP phase, the supplier must provide a design solution to win the bid. The supplier has designed and built composite flaps for OEMs, and the composite technology is considered to be applied for the rudder design. Although composite material can effectively reduce the airframe weight, it increases manufacturing cost due to high material prices and a need for labour intensive manufacturing processes. Therefore, the DOC is chosen as a figure of merit which is required to evaluate a design solution in terms of both cost and weight.

6.4.1 Description of the application case

The shape and main dimensions of the rudder used for this application case are shown in Figure 6.22. There are two hinges and two actuators by which the rudder is attached to the vertical tail. The structure is made of thermoplastic polyphenylene sulfide (PPS) material, assembled as a closed box using an induction welding technique.

The rudder is subjected to three basic types of loads: the aerodynamic pressure loads on the skins, the imposed hinge displacements loads at hinges, and the concentrated forces on jammed actuators if jamming happens. A total of 13 load cases are considered, combining and scaling the three basic loads via the input parameter *load-case-input*.

The design variables for the rudder and their bounds are shown in Table 6.7. The upper limits of the numbers of spars and ribs were given, considering the difficulty to assemble many parts together

within the small rudder OML. The bounds of the position of the front and rear spars and the rib number were suggested by engineers from the rudder supplier based on their experience. The rib and spar numbers are integer variables, whereas the rib and spar positions are continuous variables. The OML and positions of hinges and actuators were fixed during the optimization process. The ribs were evenly placed between reinforced ribs to which a hinge or an actuator was connected, whereas spars were evenly placed between the most forward and afterward position w.r.t. the root chord. The material properties of the rudder are listed in Table 6.8. The parameters for estimating labour time were adopted from Bao et.al (2000) and are shown in Table 6.9. All these parameters were specified as input parameters for the movable MMG.

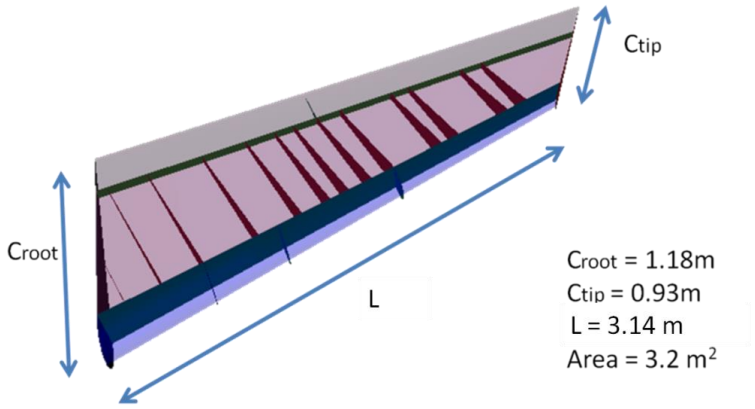


Figure 6.22: Dimensions of the rudder

Table 6.7: Definition of design variables for the rudder

Variables	Lower limit	Upper limit	Type	Description
x1	2	4	Integer	Spar number
x2	13	17	Integer	Rib number, not including end ribs
x3	0.2	0.3	Continuous	The most forward spar position w.r.t root chord
x4	0.7	0.8	Continuous	The most afterward spar position w.r.t root chord

Table 6.8: Material properties and input parameters for cost estimation for the rudder

E11	90 GPa	ρ	1600 kg/m^3
E22	9.8 GPa	P_l	0.3 dollar/min
G12	3.5 Gpa	*sr*	0.5
v12	0.29	P_m	50 dollar/cm^2

Table 6.9: Basis values for cost estimation of the rudder

Item	ν_0 [cm^2/min]	τ_0 [min]	λ
WB/LE/TE rib	5.312	10.356	Area [cm^2]
Spar	9.343	6.279	Area [cm^2]
Skin	13.833	4.388	Area [cm^2]
Assembly	0.072	2.988	Assembly perimeter [cm]

6.4.2 Design of experiment and response surfaces

The computational time required to analyse each topology was in the order of 10 minutes, using a computer with a 2.66Ghz Intel Core2Qaud Q8400 processor and 4Gb RAM memory. This included the time taken to instantiate the parametric model of the business jet rudder, to prepare inputs for cost and weight analysis, to perform FEA and composite plate sizing, and to carry out cost estimation. To avoid excessive calculation time, it was decided to make use of response surfaces. The Latin hypercube method was used to generate the sample design points for the DoE. Sixty sample designs, see Appendix I, required to generate and the cost and weight of each sample design were evaluated. The total time for generating the DoE was about 10 hours. The results of the DoE are shown in Appendix I. No manual activities were required for evaluating each sample point during the DoE.

The Kriging method was used to generate the response surfaces based on the DoE results. A MATLAB toolbox called "DACE" was used to build the Kriging model. In order to verify the quality of the response surface model, 10 random design points were generated. A comparison of values estimated via the response surfaces and the ones directly evaluated from the ADEE is shown in Figure 6.23. The average approximation error for cost was 8.7% while the error for weight was 3.2%. A Pareto front based on the Kriging model is shown in Figure 6.24. Four typical optimal designs on the front are highlighted in Figure 6.23 and will be discussed in detail in the next section.

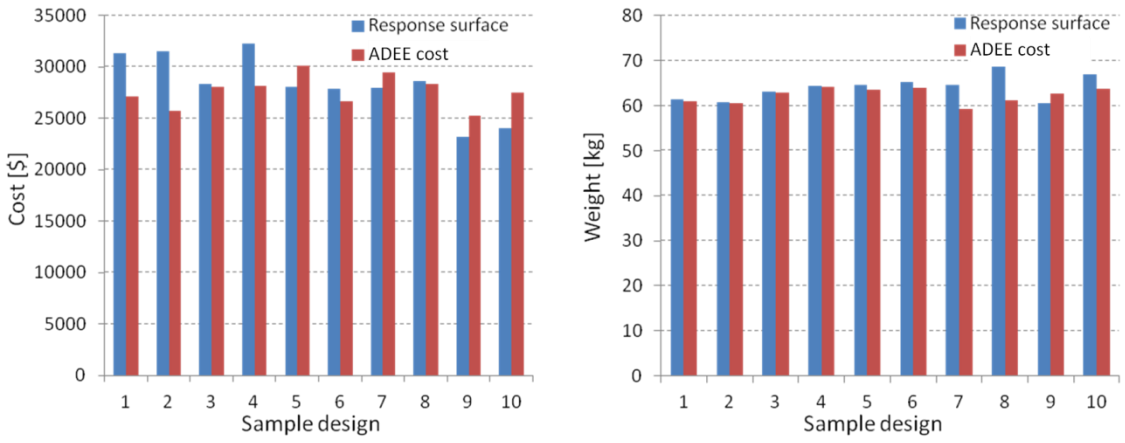


Figure 6.23: The validation results for the response surface of the rudder

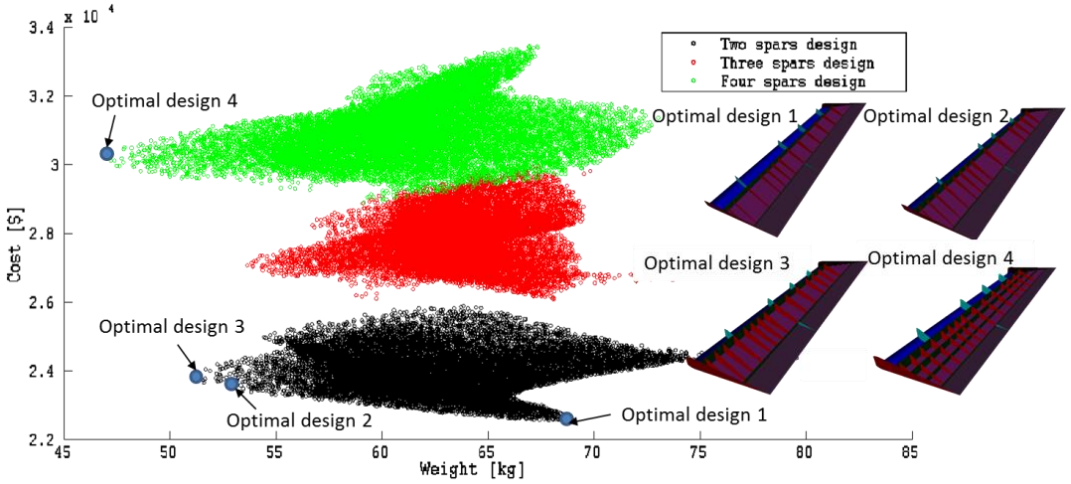


Figure 6.24: Pareto front for cost and weight of the rudder

6.4.3 Optimization results and discussions

The optimal topology varied as the value of weight penalty, see Eq. 6.1, increased from 1 to 100000 dollar/kg, as shown in Figure 6.25. The supplier could choose the best topology with a given weight penalty value from OEM, however, the method for determining the weight penalty value fell outside the scope of this research. Four typical optimal topologies on the Pareto front were found by increasing the weight penalty value, as shown in Table 6.10. These topologies are the four optimal design marked in Figure 6.24. The optimal design 1 with the lowest P value, 21, was a high-weight and low-cost design whilst the optimal design 4 with the highest P value, 1,350, was a low-weight and high-cost design. Optimum designs 2 and 3 were the compromised designs.

The minimum manufacturing cost can be found in optimal design 1 while the minimum weight design is achieved in optimal design 4. The optimal design 1 had the minimum spar and rib numbers, leading to the minimum cost; however, its weight hits the highest value because more material is required to prevent the WB skin from buckling with the largest rib spacing. The optimizer reduced the WB skin area and rib areas by moving the front spar towards TE and the rear spar towards LE.

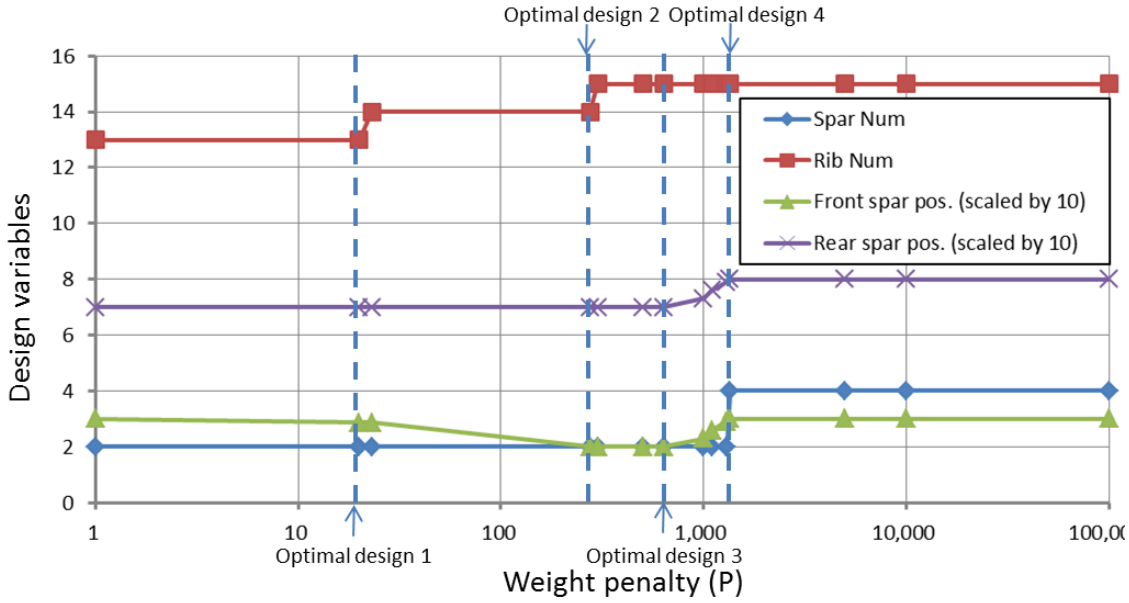


Figure 6.25: Optimal design influenced by the weight penalty value

Table 6.10: The optimization results of the four optimal designs

Optimal design	*Spar no.*	*Rib no.*	*Front spar Pos.*	*Rear spar pos.*	*cost ($)*	*Weight (kg)*	*P ($/kg)*
1	2	13	0.29	0.7	22,532	68.3	21
2	2	14	0.2	0.7	23,471	52.7	275
3	2	15	0.2	0.7	23,820	51.1	637
4	4	15	0.3	0.8	30,198	46.5	1,350

The optimal design 4 had the maximum spar number which reduced the width of the WB skin panels, resulting in a smaller thickness of the WB skin than the LE/ TE skin. It was interesting to note that the front spar was moved as far afterwards as possible. Part of the reason might be that moving the front spar back effectively reduced the area of the WB ribs, yielding less WB rib weight. The rib number for this optimal topology was 15 instead of the maximum allowed rib number 17. This implies that if one puts an additional rib into the optimal topology, the weight saving due to smaller rib spacing will not outweigh the weight of the additional rib. The optimal design 4 yielded the highest cost value, 30198 dollars, because the four-spar design had the most assembly joints between the ribs and spars, leading to the highest assembly costs which accounted for more than 60% of the total cost, as shown in Figure 6.26; and therefore, the highest manufacturing cost.

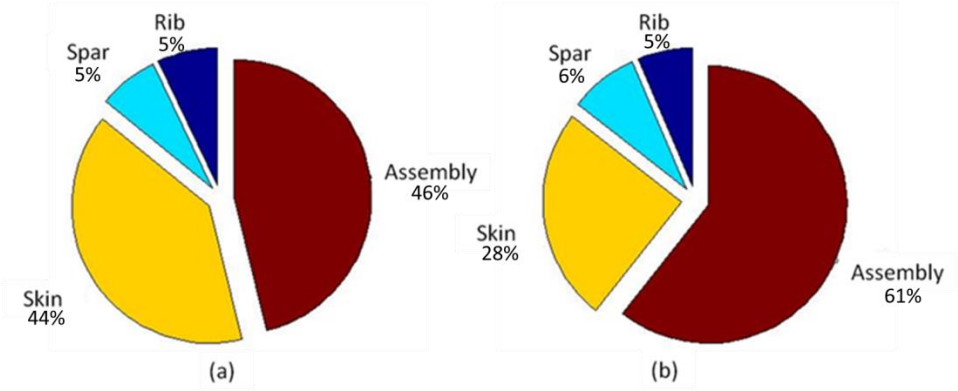


Figure 6.26: (a) Cost breakdown for optimal rudder design 1; (b) cost breakdown for optimal rudder design 4, where assembly cost accounts for more than 60% of the total cost

6.5 Concluding remarks

The movable ADEE was successfully applied to the design of a composite rudder, for which the Pareto optimal set of structural configurations for minimum weight and manufacturing cost had been identified. Using this Pareto set, the rudder manufacturer could conduct internal trade-off studies between minimum weight and minimum cost solutions, and offer the OEM a broad set of optimized options, rather than just one feasible design.

As a demonstration system for the global-local knowledge coupling, the movable ADEE meets the requirements defined in Section 3.4.3. The movable MMG is flexible enough to model movables of different configurations, as shown in Figure 6.12. To perform the DoE for optimizing the business jet rudder, sixty different topologies of internal structures have been instantiated by the MMG. The ADEE integrates a FEA-based weight estimation method and a bottom-up parametric cost estimation method. Using the DoE results, it can be seen that the weight estimation method and the cost estimation method are sensitive to topology changes in the internal structures.

The repetitive activities in the optimization process are automated by capturing the local rudder design knowledge. The *PCL-writer* CM, *Surface-splitter* CM and *Smart-data-collector* CM capture the process knowledge required for pre-processing/post-processing of the movable FE model. The *DV-region-data-collector* CM captures the process knowledge required to prepare inputs for sizing the structural components. The *Weight-input-collector* CM and *Cost-input-collector* CM capture the process knowledge for preparing inputs for weight estimation and cost estimation.

Chapter 7. Conclusions and future work

The objective of this research discussed here was to *develop a design approach which can support suppliers to quickly perform airframe local designs from which critical results, i.e. cost and weight, can be generated in the overall aircraft conceptual design phase.* The conclusions and future work are discussed in this chapter.

7.1 Conclusions

The outsourcing that has taken place in the aircraft industry over the last few decades has created a globalized supply chain for a limited number of OEMs. Multi-level design has resulted in response to a shift in airframe subsystem design to suppliers. The OEM focuses increasingly on the requirement allocation and definition for airframe subsystems and verification at a global level, whereas suppliers focus on the realization and improvement of airframe subsystems at a local level.

Relying on the supply chain for innovative design and build can land an OEM in the situation that it has insufficient bottom-up knowledge about local design, in particular, innovative local designs, e.g. composites and new production methods. Suppliers who have detail-level knowledge should be involved early in the overall aircraft conceptual design phase by creating various local designs, and by conducting more accurate analysis and evaluation of these designs.

The global-local knowledge coupling supports local design using capture of global design knowledge and automation of the repetitive design tasks. It comprises two modules at the global and local design levels respectively: one is the cross-over, which captures global design knowledge and provides the input required for starting a local design. The module at local design level is a set of parametric product and process models, of airframe subsystems, used to automate repetitive design actions at local design level, such that the analysis and evaluation of subsystem designs can be quickly performed.

The design engineering and engine, a KBE inclusive MDO framework concept, was chosen to embody the proposed approach. Three instantiations of the DEE (van Tooren, 2003), namely the fuselage ADEE, fuselage panel ADEE and movable ADEE, were developed to demonstrate the functionalities of the global-local knowledge coupling.

The cost and weight estimation methods integrated in the ADEEs are bottom-up and physics-based. The airframe cost is predicted based on a bottom-up parametric cost estimation method adopted from the literature (van der Laan, 2008). The method links the cost data with the characteristics of airframe structural members, therefore it allows designers to search for a cost effective design. The weight estimation method uses finite element analysis to calculate the internal stresses of the structural elements and an analytical sizing method to determine their minimum required thicknesses. Both the cost and weight estimation methods are design sensitive to the characteristics of the airframe structural members.

In Chapter 4, the fuselage ADEE was implemented as a cross-over which can provide load sets and fuselage OML for the local panel design. The global knowledge was captured in the cross-over, including the knowledge of how to generate fuselage OML and the knowledge of how to perform disciplinary analysis such as load calculation and structural analysis using FEA. The fuselage ADEE

can also be used by panel suppliers as a stand-alone tool to capture the effects of material and structural layout on fuselage weight.

In Chapter 5, the fuselage panel ADEE was implemented to demonstrate the high performance of the local panel design. The fuselage ADEE is a cross-over which provides inputs for this local design. A KBE-enabled parametric model of a panel was developed within the fuselage panel ADEE, including skin with multiple layers, and back-up structural members, such as frames and stringers. The local knowledge was captured in the panel ADEE to automate the panel modeling, global-local FEA modelling for structural analysis, parametric bottom-up cost estimation and weight evaluation.

In Chapter 6, the movable ADEE was developed to demonstrate the optimization capability of the global-local knowledge coupling. Receiving the movable OML and load sets from OEM, the movable ADEE is used to perform cost/weight multi-objective optimization of movable structures, including large topology variations of the structural configuration. The capability of the framework was successfully demonstrated by designing and optimizing the composite structure of a business jet rudder. This study case shows that this ADEE is able to find the Pareto optimal set for minimum structural weight and manufacturing cost in ten hours using a computer with a 2.66Ghz Intel Core2Qaud Q8400 processor and 4Gb RAM memory.

The KBE-enabled modelling tools (MMGs) of the DEE concept guarantee the model consistency between global design and local design, and keep the consistency between disciplines. The analysis models used for structural analysis and cost estimation are derived from the same parametric and generative model for each ADEE.

The DEE concept is modular and the modules of these ADEEs can be reused for future projects. For example, the capability module *PCL-writer* for automatic FE model generation was successfully used for all of the ADEEs presented here, and it can be reused for other subsystems such as the wing box. The composite plate sizing method can be reused for sizing the structural members of other composite airframe subsystems such as a composite fuselage.

The cross-over of the global-local coupling can reduce the local design's dependence on global design, and hence the design process of local design can be shifted forward. Supported by KBE, the local design and the cross-over of the global-local coupling can generate and analyse various design variants within a short lead time by automating model (re)generation and preprocessing for discipline analysis. Therefore, it can be stated that the global-local coupling can support suppliers in performing airframe local designs quickly from which critical results, i.e. cost and weight, can be generated in the overall aircraft conceptual design phase.

7.2 Limitations and Future work

A limitation lies in the implementation of the cross-over within the suppliers' domain. Suppliers do not have the in-depth knowledge in the overall aircraft design as OEMs although suppliers have in-depth knowledge about the airframe subsystems. The quality of the inputs needed to start the local design is reduced when the over-simplified methods are used within the cross-over. A risk arises that the optimization of the local design might be meaningless, relying on the low-quality inputs provided by the cross-over. Another limitation is that there is no feedback from the local design module to the cross-over. For example, the fuselage panel stiffness from the local panel design may be no longer consistent with the global model used within the cross-over. The global model should adapt to the

stiffness changes from the local level. Then, a new panel design should be defined, and cost and weight should be re-estimated/evaluated, the panel should be re-analyzed structurally, and so forth, until the panel stiffnesses of the global and local models converge.

There are many possibilities to improve the acceptance level of these demonstration design systems that have been developed so far. One is to use more realistic analysis tools in the ADEE. Manufacturability should be analyzed because it indicates whether a part/component can be manufactured using a certain material and a certain manufacturing technique. Further development of the ADEEs presented here is necessary to support common structural concepts, such as the sandwich skin and stiffened panel, and more realistic cost estimation tools from industry should be included in the ADEEs.

Another future improvement would be to develop a more sophisticated MMG that can model more type of structural members. The bottom-up approach used for cost and weight estimation needs to reflect the realistic structural complexity of airframe systems in their product models. In this research, certain levels of simplification were made for the MMGs in all three ADEEs since the ADEEs were prototype systems used to verify the design approach.

Surrogate models of disciplinary analysis should be built in the future to relate the important behaviours of local design, i.e. cost, weight and stiffness, with more design variables. These surrogate models can provide both suppliers and OEM with insights with respect to trade-off between disciplines.

Last but not least, user-friendly graphical interfaces need to be developed for these tools to make them convenient and attractive to use for a wider group of users.

Appendix A The PCL-writer

A.1 Three types of PCL files for automated FEM generation

Three types of files are needed for automated FEM generation. One, the main execution files which define the execution process of all the other PCL files, such as "pre.pcl" and "post.pcl". Two, the files which define static PCL functions. The static PCL functions are a self-contained program unit consisting of built-in PCL statements. They are used to break the execution process into logical modules and pass arguments back and forth to other PCL functions or to the main program. Three, the PCL files generated by the *PCL-writer* which contains the customized commands calling the built-in PCL functions or calling the static PCL function defined in the second type of PCL files. The first and second types of PCL files are static files, which are kept the same for all the topologies of a structure. The third type of PCL files is dynamic and the commands within them change for different topologies to be analyzed. The third type of PCL files is generated by the LISP functions described in Section A.3.

A.2 Different numbering methods between GDL and PATRAN

As discussed before, the parametric model in this research is surface-based. This means that most of the geometric entities imported into PATRAN are surfaces, called here segments. The loads and boundary conditions (BCs) of a FE model are often defined and associated with the vertices or the edges of these surfaces in GDL.

The GDL and PATRAN use different numbering methods, for example, the 1st edge of the quadrilateral surface in GDL is the 3rd edge of the surface in PATRAN. This difference will cause difficulty when applying loads and BCs to the right geometric entities.

A mapping pattern was observed to keep the traceability of vertices and edges between GDL and PATRAN. This pattern is shown in Figure A.1.

To refer to an edge in PATRAN, two integers are required, for example, to refer to the 2nd edge of quadrilateral surface 101, a string "surface 101.2" is used and the two integers are 101 and 2. It is very important to point out that "surface 101.2" is the 4th edge of segment 101 in GDL. Three integers have to be specified to refer to a vertex of a surface. For example, to refer to the 2nd vertex of surface 101, a string "surface 101.2.1" is used to select the vertex and 101, 2 and 1 are included in the string.

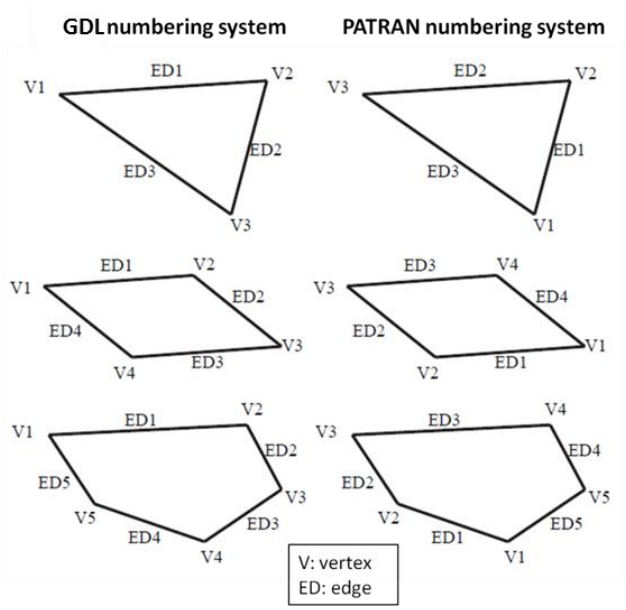


Figure A.1: An example of beam-properties-data-list

A.3 LISP functions of the PCL-writer

The LIPS functions of the *PCL-writer* generate the PCL file to steer the pre-processing and post-processing of the FE model. There are string templates of the particular commands in PATRAN. Before giving code examples of the *PCL-writer*, the LISP terminology will be given first.

The acronym LISP is an abbreviation of List for Processing. Everything in LISP is a list which is enclosed by two parentheses, e.g.

(0 1 2 3), ("A" "B" "C" "D") or (object1 object2 object3 object4)

These lists have four elements. A Plist is a list containing keyword-value pairs. An example of a Plist named "A" is:

(:keyword-1 value-1 :keyword-2 value-2)

A Plist may have a nested structure. This means the value of a keyword-value pair can be another Plist. For example, if value-1 and value-2 are (:keyword-3 value-3 :keyword-4 value-4) and (:keyword-5 value-5 :keyword-6 value-6) respectively, the Plist A turns to be

(:keyword-1
(:keyword-3 value-3 :keyword-4 value-4)
:keyword-2
(:keyword-5 value-5 :keyword-6 value-6))

Plists can be elements of a list, e.g.

((:keyword-1 value-1 :keyword-2 value-2) (:keyword-3 value-3 :keyword-4 value-4))

Most input arguments of the LISP functions of the PCL-writer are the lists whose elements are a (nested) Plist. The most common LISP functions will be described below.

beam-property-pcl-writer(beam-properties-data-list)

This function generates two PCL files which are used to create beam elements in PATRAN. An example of a beam-properties-data-list is shown in Figure A.2.

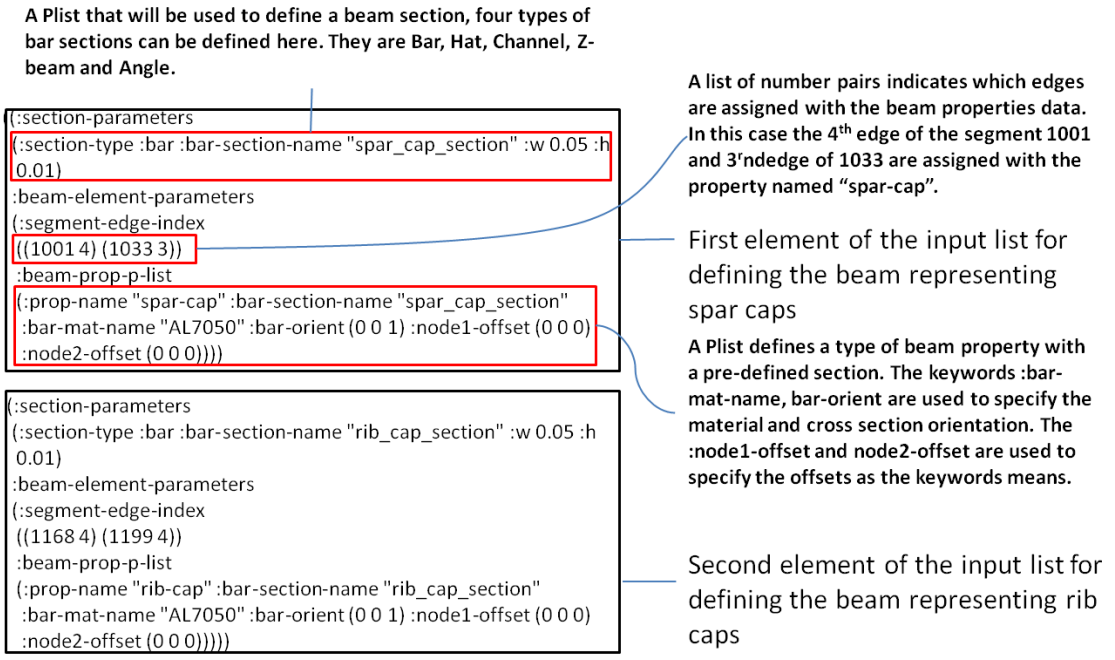


Figure A.2: An example of beam-properties-data-list

The "beam_prop.pcl" is used to create the beam properties in PATRAN while the "create_beam_elements.pcl" will be used to create beam elements with the created beam properties. The content of these files are shown in Figure A.3.

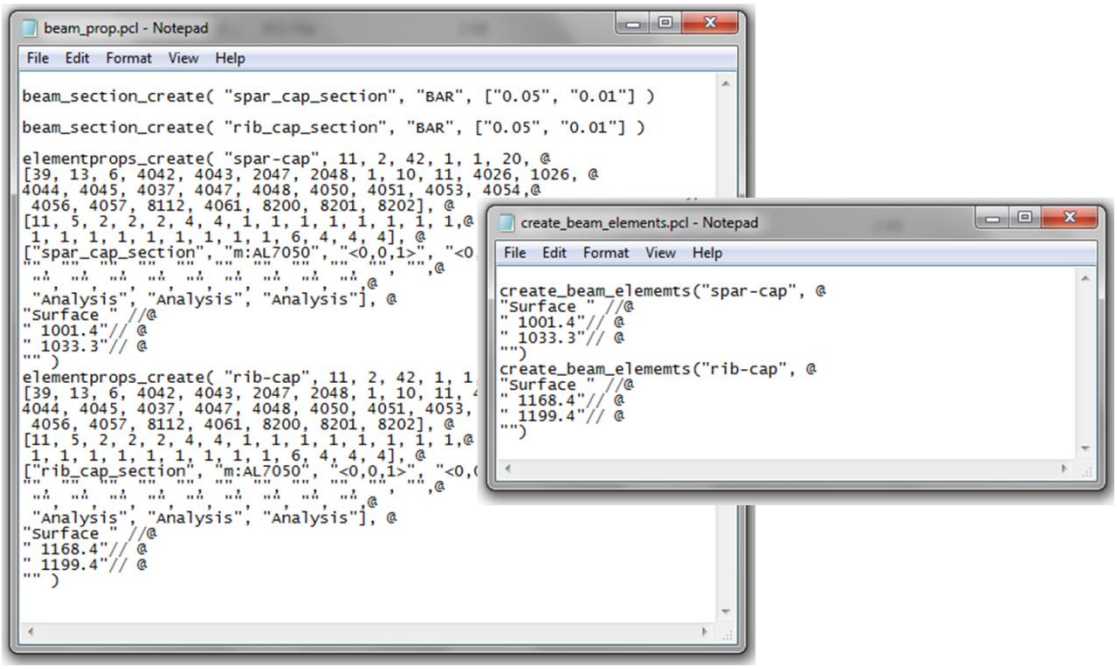


Figure A.3: The PCL files generated by beam-property-pcl-writer

shell-property-pcl-writer(cad-file-name segment-properties-data-list)

The segment-properties-data-list is a list in which the properties of all the segments should be defined. The i^{th} element of segment-properties-data-list is a Plist which specifies the property data of the i^{th}

segment in the segment-list. Therefore, the element number of the list is the same as the number of the segments. An example of the segment-properties-data-list is shown in Figure A.6. In this example, the segment-properties-data-list only has two elements, meaning only two segments are generated from the segmentation process. The first segment is a monolithic or aluminum plate while the second segment is a composite plate. An example of a segment-properties-data-list is shown in Figure A.4.

Five PCL files are generated when the shell-property-pcl-writer function is called. The file "import.pcl" is generated to import the STEP file containing two segments. Meantime, the segments will be renumbered according to the user-defined IDs. The file "group.pcl" is used to group the segments. For example a rib name "Rib1" has four segments. The value of :group-name is set as "Rib1" for all these segments. After executing "group.pcl", these segments will be grouped into the same "Rib1" group. The file "ms.pcl" is used to seed all the edges of the segments. A global parameter *mesh-seed-num* is used to control the seed number on all the segment edges. The file "shell_prop.pcl" is used to create the shell properties in PATRAN while the file "create_shell_elements.pcl" is used to create shell elements with the created shell properties. The "Isomesh" mesher is chosen for a segment with four edges while the "Paver" mesher for a segment has two, three or five edges.

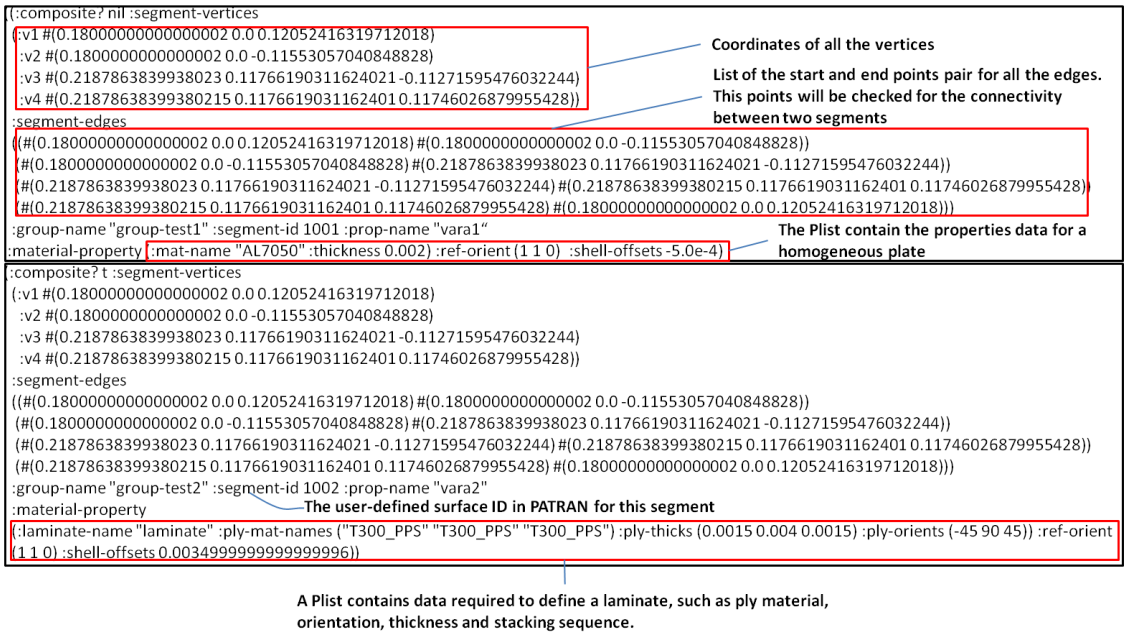


Figure A.4: An example of a segment-properties-data-list

coord-pcl-writer(pcl-file-name coord-input-data)

This function is used to generate a PCL file for defining coordinate frames. An example of coord-input-data is shown in Figure A.5.

```
((:coord-id 1 :coord-orign "[0.0,0.0,0.0]" :point-on-axis1 "[0.0,0.0,-0.3]" :point-on-axis2 "[-0.28,0.10,0.0]")
 (:coord-id 2 :coord-orign "[0.3,0.3,0.]" :point-on-axis1 "[0.5,0.3,0.0]" :point-on-axis2 "[0.3,0.5,0.0]")
 (:coord-id 3 :coord-orign "[0.5,0.5,0.]" :point-on-axis1 "[0.7,0.5,0.0]" :point-on-axis2 "[0.5,0.7,0.0]"))
```

Figure A.5: An example of coord-input-data

geometry-bc-pcl-writer(pcl-file-name geometry-bc-input-data)

The function generates a PCL file which is used to a displacement boundary condition associated with geometric entities. The boundary condition indicates which degrees of freedom of the specified **geometric entities** are constrained. An example of **geometry-bc-input-data** is shown in Figure A.6.

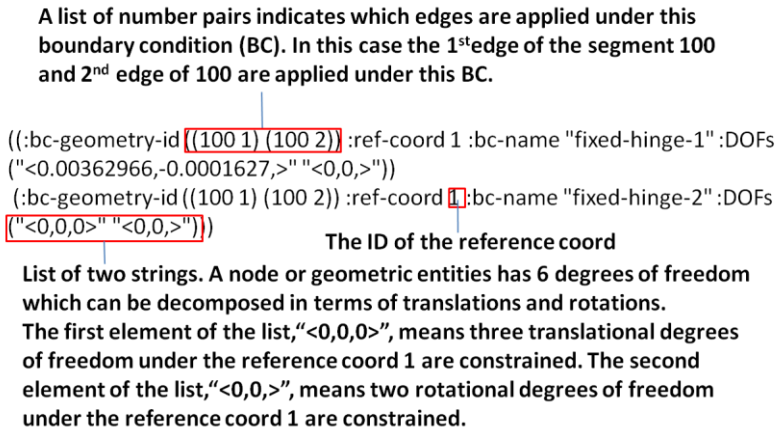


Figure A.6: An example of geometry-bc-input-data

node-bc-pcl-writer(pcl-file-name node-bc-input-data)

The function generates a PCL file for creating a nodal displacement boundary condition. The boundary condition indicates which degrees of freedom of the specified **node(s)** are constrained. An example of the inputs for this function is shown in Figure A.7.

```
((:bc-node-id (101 102) :ref-coord 1 :bc-name "fixed-hinge-1" :DOFs
("<0.00362966,-0.0001627,>" "<0,0,>"))
 (:bc-node-id (1001 1002) :ref-coord 1 :bc-name "fixed-hinge-2" :DOFs
("<0,0,0>" "<0,0,>")))
```

Figure A.7: An example of node-bc-input-data

pressure-field-pcl-writer(pcl-file-name field-input-data)

This function is used to generate a PCL file for defining the spatial fields in PATRAN. An example input of filed-input-data is shown in Figure A.8.

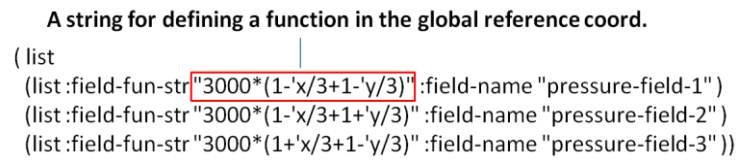


Figure A.8: An example of field-input-data

pressure-load-pcl-writer(pcl-file-name pressure-load-input-data)

This function is used to generate a PCL file for applying the pressure load with the distribution described in a spatial field on the top of the segments. An example input of filed-input-data is shown in Figure A.9.

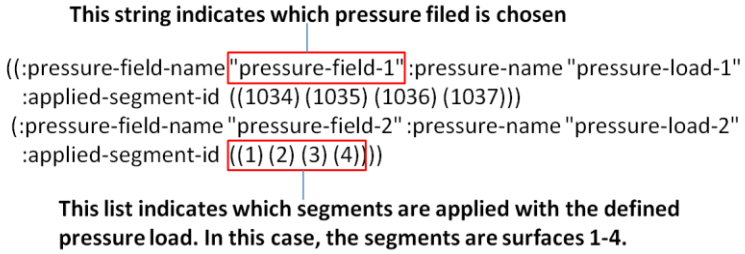


Figure A.9: An example of pressure-load-input-data

rbe2s-pcl-writer(pcl-file-name rbe2s-input-data)

Given the coordinates of independent nodes and vertexes of segments where attached to the node, the function will create RBE2 elements. RBE2 elements can be used to represent the connections, e.g. hinges and actuators. These elements might be used to assemble different components, e.g. wing and fuselage. An example input of rbe2s-input-data is shown in Figure A.10.

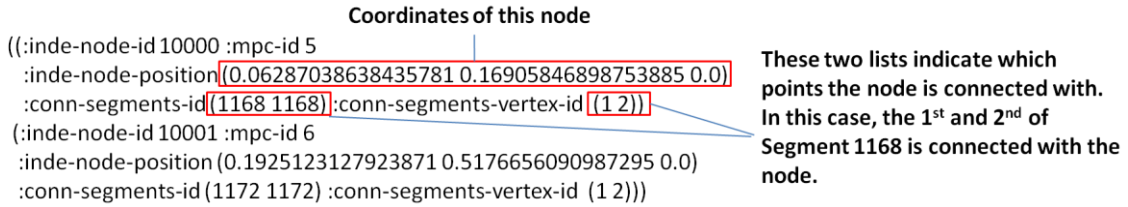


Figure A.10: An example of rbe2s-input-data

load-case-pcl-writer(pcl-file-name load-case-input-data)

This function is used to generate a PCL file for defining load cases for FEA. An example input of load-case-input-data is shown in Figure A.11.

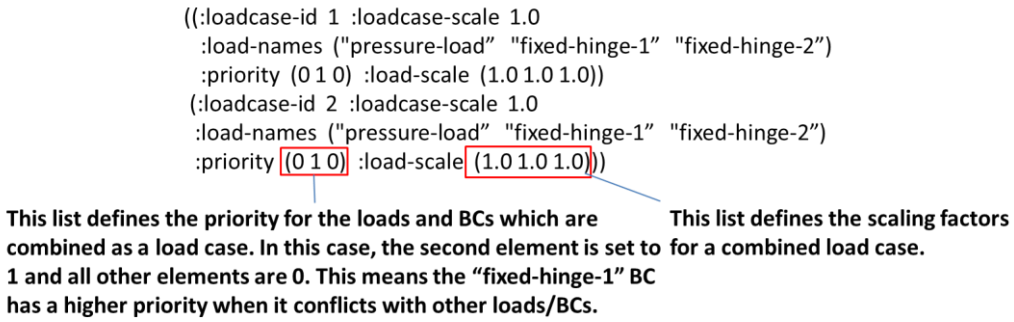


Figure A.11: An example of load-case-input-data

concentrated-load-pcl-writer(pcl-file-name load-input-data)

This function generates a PCL file for defining concentrated loads on surface vertices. An example input is shown in Figure A.12.

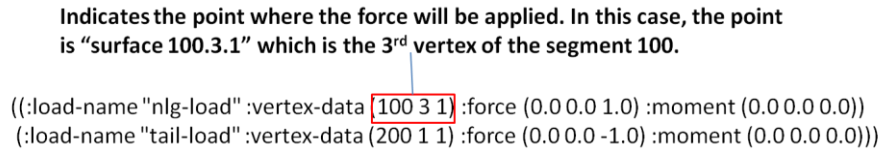


Figure A.12: An example of load-input-data

discrete-mass-geometry(node)-pcl-writer(pcl-file-name discrete-mass-input-data)

This function generates a PCL file for defining a lumped point mass in PATRAN. The mass can be attached to a geometric entity or a node. An example input of discrete-mass-input-data is shown in Figure A.13.

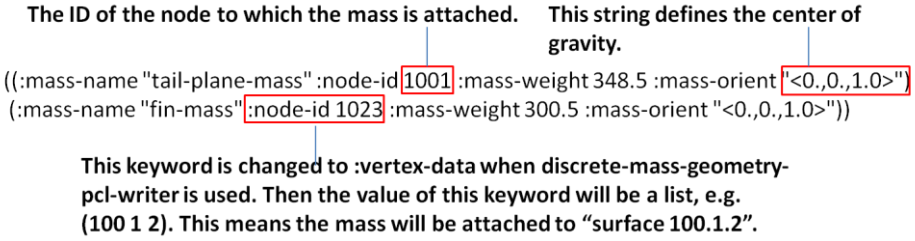


Figure A.13: An example of discrete-mass-input-data

A.4 Use of the PCL-writer

The steps required to use the *PCL-writer* are explained in this section.

Step 1. Move the source code "PCL-writer.lisp" to the "../source". All the LISP functions described in Section A.3 can be found in "PCL-writer.lisp". Set the global parameter *patran-installation-pathname*. Create a folder "../cad files" where a STEP file containing all the segments will be saved, and another folder "../output-files/PCL/static" where the static PCL files will be saved.

Step 2. Create a computed slot named as "run-PCL-writer" in the user-defined object. In this slot, the sequence of calling LISP functions of the *PCL-writer* is defined. Figure A.14 shows an example of the slot.

Step 3. Define the execution process of the PCL files in "Pre.pcl". A template of "pre.pcl" is stored in "../output-files/PCL/static". Call the computed-slots "run_preprocessing" to execute the PCL files, following the sequence defined in "Pre.pcl".

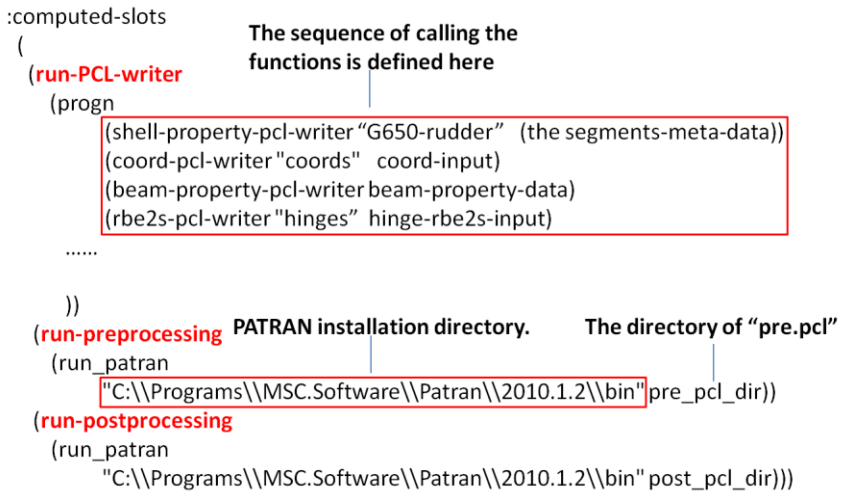


Figure A.14: Inserted computed slots

Appendix B Input parameters for the fuselage MMG

Parameter	Example	Description
Structural layout		
frame-para-list	(list (0.0 0 t) (0.6 0 t) … (41.4 0 nil)(42.0 0 t))	Longitudinal position: 0.0 0.6 41.4 42.0.Virtual/real frame: t nil. All the orientation angles of frame cut plane is 0.
pressure-bulkhead-cs-list	‘(0 43)	The front and rear pressure bulkheads are attached to the 1st and 44th frame defined in the *frame-para-list.*
main-frame-cs-list	'(0 3 18 22 26 39 44 47)	The 1st, 4th and 19th etc frames are reinforced as main frames.
floor-strut-node-index	‘(3 20)	Node-index-on-floor-beam is 3, whereas the Node-index-one frame line is 20. The floor struts are created by two control points, the 4th control point on the floor beam curve and 21st on the frame curve.
floor-strut-cs-list	‘(1 2… 16 17 27 28 … 38 39)	Define which frames are attached with floor struts.
keel-beam-width	4	The last 4 skin elements are kept as a keel beam in a frame bay
cut-out-para-list	‘(‘(:start-frame-cs-index 22 :end-frame-cs-index 26 :start-circumferential-index 16 :end-circumferential-index 26) …)	Define a rectangular area in which the elements and nodes are deleted. All the cut-out areas such as doors are defined here.
upper-center-wing-height	-0.6	The z coordinate of the top point of the front spar of center wing box if there is no center wing trunk is defined. The center wing trunk is used to define the structures of the center wing box.

lower-center-wing-height	-1.4	The z coordinate of the low point of the front spar of the center wing box if there is no center wing trunk is defined.
wb-spar-height-difference	0.15	The height different between the front spar and the rear spar if there is no center wing trunk is defined.
frame-para-list	(list (0.0 t) (0.6 t) … (41.4 nil)(42.0 t))	Longitudinal position: 0.0 0.6 41.4 42.0.Virtual/real frame: t nil
pressure-bulkhead-cs-list	‘(0 22 43)	The front and rear pressure bulkheads are attached to the 1st, 23rd and 44th frame defined in the *frame-para-list.* The pressure bulkhead which encloses the main landing gear wheel well is defined here for the low wing configuration.
pressure-bulkhead-inner-radius	'(0.2 0.2)	Specify the radius of inner boundary curve for each bulkhead.
main-frame-cs-list	'(0 3 18 22 26 39 44 47)	The 1st, 4th and 19th etc frames are reinforced as main frames.
floor-strut-node-index	‘(3 20)	The floor struts are created by two nodes, the 4th nodes on the floor beam curve and 21st on the frame curve.
floor-strut-cs-list	‘(1 2… 16 17 27 28 … 38 39)	Define which frames are attached with floor struts.
wb-root-airfoil	“naca0006”	The root airfoil type of center wing trunk.
wb-tip-airfoil	“naca0006”	The tip airfoil type of center wing box.
wb-position	‘(20 0 0.5)	The location of the root of the leading edge of the center wing box.
wb-width	4	The width of the center wing trunk.
wb-length	1.5	The length of the center wing trunk.

front-spar-location	0.25	The front spar is straight and locates at the 25% of the root chord of the center wing trunk.
rear-spar-location	0.75	The front spar is straight and locates at the 75% of the root chord of the center wing trunk.
rib-location	‘(0 0.1 0.2 … 0.9 1)	The ribs are all perpendicular to the front spar.
Materials & Sections		
structural-member-material-input	‘ (:frame ‘AL2024’ :floor-beam ‘AL2024’ :CWB-pressure-bulkhead-web ‘AL7075’ …)	Define materials of structural members except fuselage skin. The material of fuselage skin is defined in skin-material-input The material mechanical properties are defined in a static PATRAN session file.
panel-material-input	‘(:mat-name ‘AL2024’ :sigT_yield 320e6 : sigc_yield 220e6 :tau_yield 285e6 :sig_hoop 92e6 :sigC_crack 180e6 :E 73e9 :rho 2700 :nu 0.318)	Define the mechanical properties for the fuselage panel material. The stringer and skin is made of the same material in the current panel sizing tool.
panel-stringer -type	:Z	Define the panel stringer type. Currently, Zee type stringers are supported in the panel sizing tool.
general-section-input	‘(‘(:section-type :z :bar-section-name “frame” :w 0.08 :t0 0.004 :h1 0.1 :h2 0.108 :offset 0.053)…)	Define section parameters for the beam-like structures. section-type has :z :hat :bar :c four options. w, t0, h1 and h2 are dimension parameters. :offset is used to offset the section profile w.r.t the lofted line.
Meshing parameters		
floor-surf-ms-num	5	The long edges of the floor surface between two frames are evenly meshed into five parts.
upper-fuse-surf-ms-list	‘(0 0.1 0.2 … 0.9 1)	The frame curve above the floor is meshed according to the

		parameters.
lower-fuse-surf-ms-list	‘(0 0.1 0.2 … 0.9 1)	The frame curve above the floor is meshed according to the parameters.
Maneuvering input parameters		
flight- altitudes	‘(10000 10000 6400)	Define flight altitudes under load cases 1-3.
mach-numbers	‘(0.85 0.85 0.85)	Define the mach numbers under load case 1-3.
weights-under-load-cases	‘(236160 236160 236160)	Define the aircraft weight under load cases 1-3.
Convergence tolerance		
weight-convergence-tolerance	50	The estimation process stops when the change of weight estimation value is less than 50 kg.

Appendix C Validation of the fuselage panel sizing module (Zee type stringer)

Table C.1: Test optimization start points

Start point	tsk	tstr	a	h	bs
1	3	3	50	50	180
2	1.5	2	40	40	220

Units: mm

Table C.2: Test design loads

Loads	Long. tension	Circum. tension	Shear	Compression	Circum. tension (fatigue)	Circum. tension (residual)
1	100	140	50	-100	90	90
2	250	140	100	-300	90	90

Units: N/mm

Table C.3: Material mechanical properties

Material	Yield stress [Mpa]	Shear yield [Mpa]	Fatigue limit [Mpa]	Residual strength [Mpa]	Young's modulus [Gpa]	Density [Kg/m^3]
AL 2024	320	285	91	69	73	2700

Table C.4: Design variables bounds

Variables	tsk	tstr	a	h	bs
Low bounds	0.5	0.5	33	33	120
Up bounds	10	10	100	100	240

Units: mm

Table C.5: Optimization results

Test case	Opt. weight	tsk	tstr	a	h	bs
1 (Start point1, load 1)	6.9004	2.1	0.5	33	38	120
2 (Start point2, load 1)	6.9004	2.1	0.5	33	38	120
3 (Start point1, load 2)	8.5231	2.7	0.5	33	38	120
4 (Start point2, load 2)	8.5231	2.7	0.5	33	38	120

Units: mm

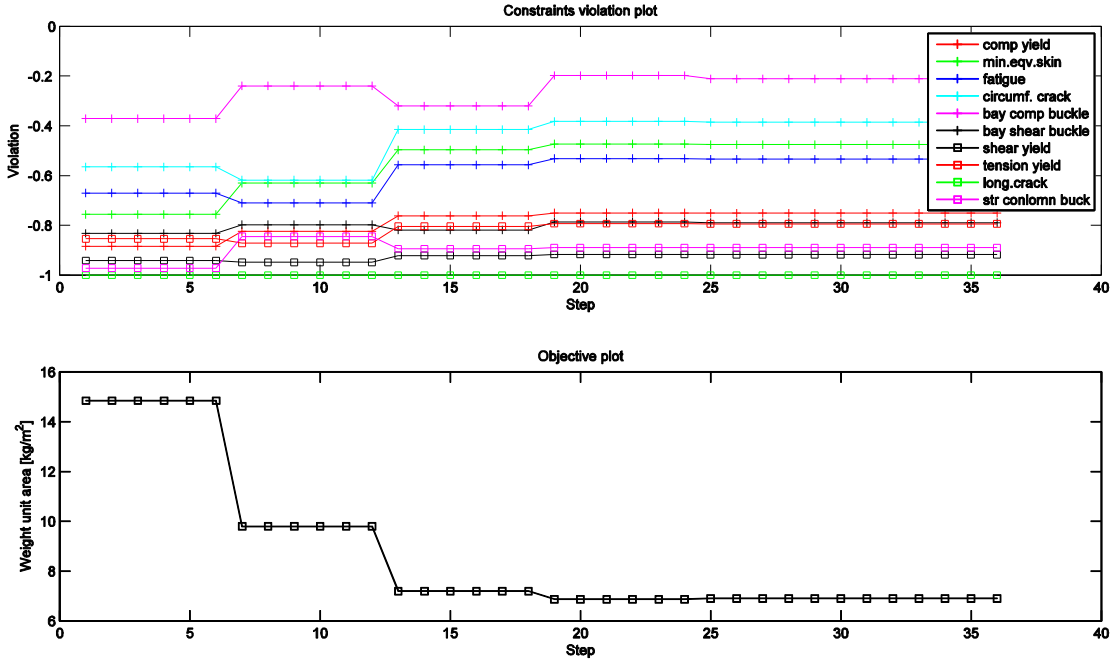


Figure C.1: Convergence history of test case 1

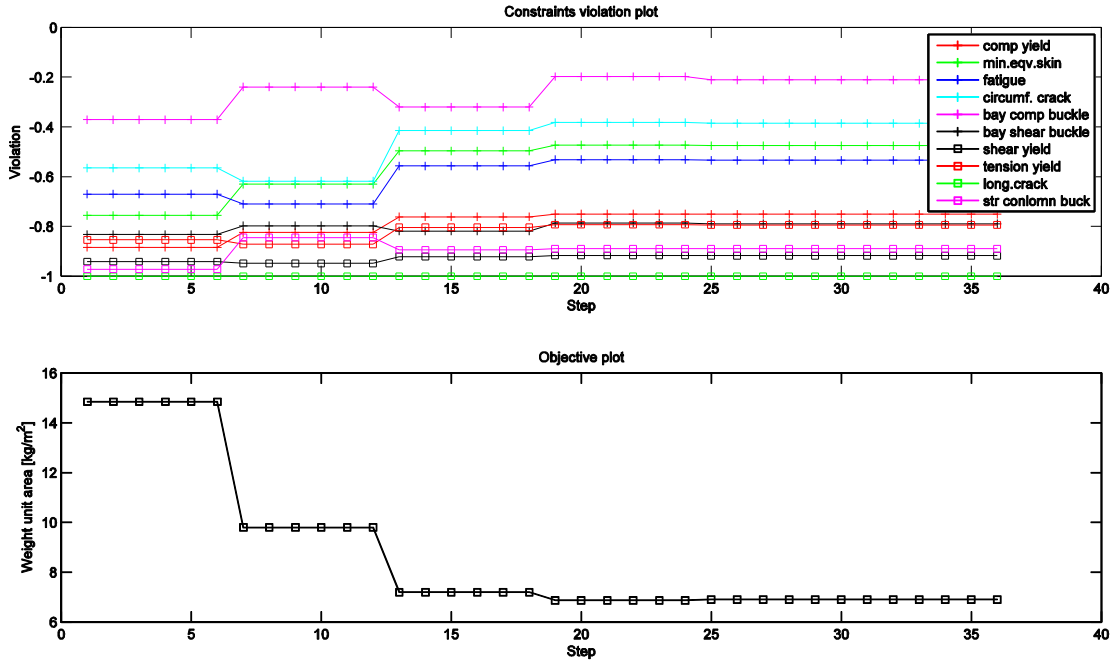


Figure C.2: Convergence history of test case 2

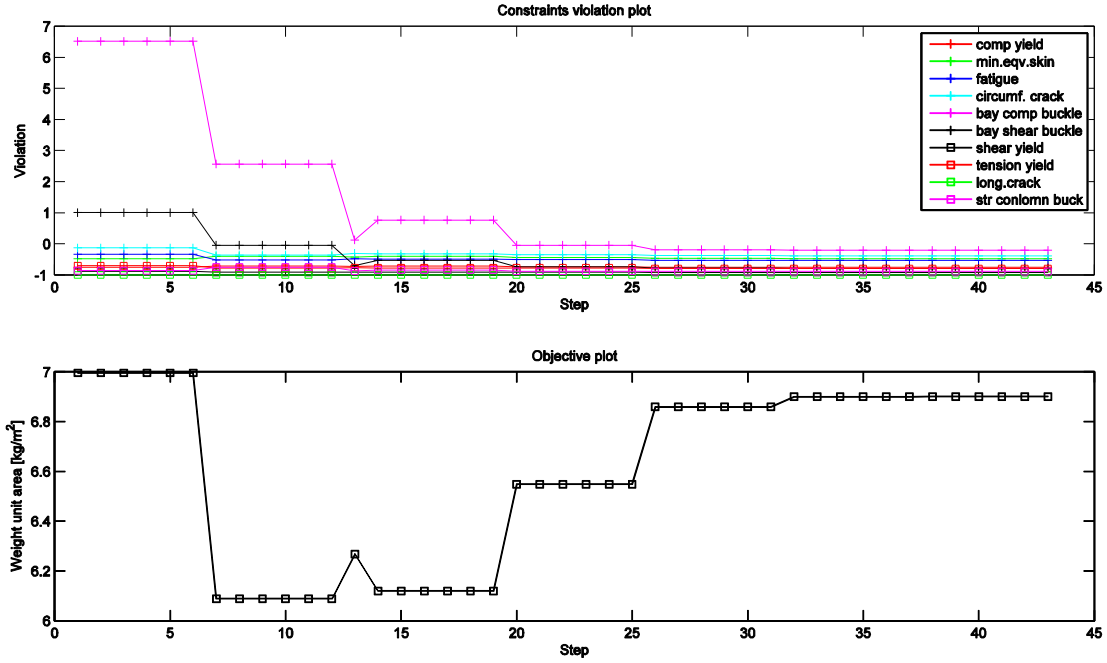


Figure C.3: Convergence history of test case 3

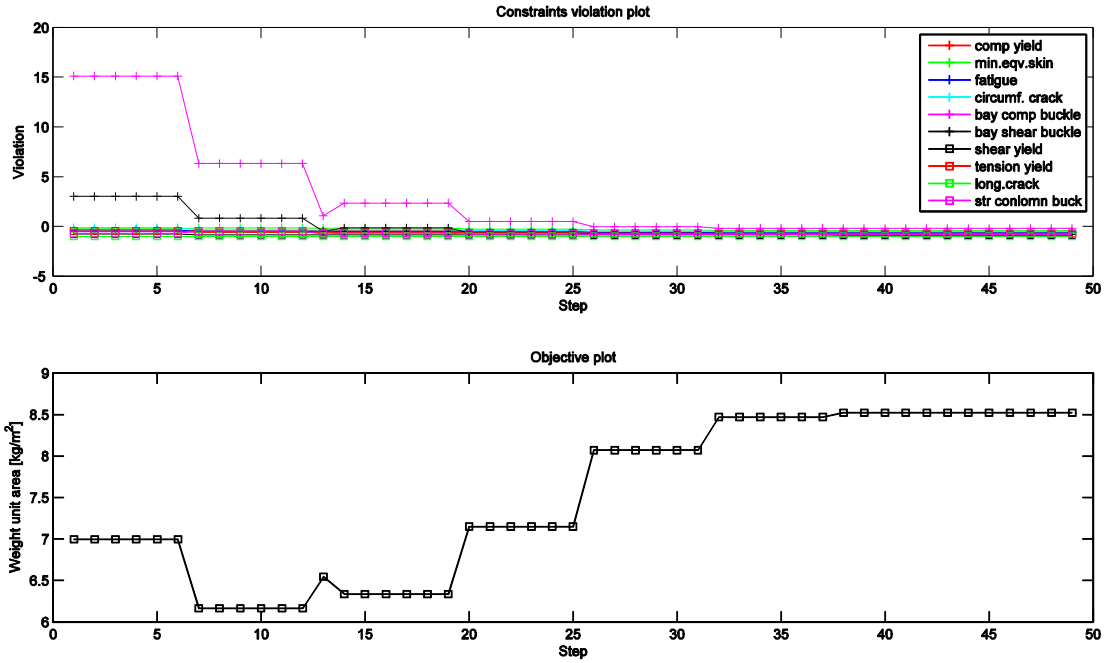


Figure C.4: Convergence history of test case 4

Appendix D Empirical equations for fuselage weight estimation

Equations for Torenbeek Class II weight estimation (1982)

Torenbeek Class II weight estimation for Al-alloy fuselages (Torenbeek, 1982)

$$W_{fuse} = (k_1 + k_2 + k_3 + k_4) * 0.23 * {S_f}^{1.2} * \sqrt{\frac{V_d * l_t}{(b_f + h_f)}} \tag{D.1}$$

where b_f, h_f are the width and height of fuselage cabin respectively; S_f is the fuselage wetted area; V_d is the design dive speed; l_t is the distance between the aircraft center gravity with the aerodynamic center of the tailplane; k_1-k_4 are the weight penalty coefficients which can be determined as follows

$k_1 = 1.08$ for pressurized fuselage; otherwise $k_1 = 1$.

$k_2 = 1.07$ for fuselage-attached main landing gears; otherwise $k_2 = 1$.

$k_3 = 1.04$ for fuselage-mounted engines; otherwise $k_3 = 1$.

$k_4 = 1.1$ for freight airplanes; otherwise $k_4 = 1$.

Equations for Torenbeek Class II weight estimation (2013)

Torenbeek Updated Class II weight estimation for Al-alloy fuselages (Torenbeek, 2013)

$$\begin{gathered} W_{fuse} = 60 * {d_f}^2 * \left(l_f + 1.5\right) * \mathrm{n}^{0.5} * d_f * l_f \\ d_f = 0.5 * (b_f + h_f) \end{gathered} \tag{D.2}$$

where b_f is the width of the fuselage cabin, and h_f is the height of the fuselage cabin.

Equations for calculating weight penalties and other fuselage structures

The skin and stringer weight are calculated based on the FE method. The weight penalties and weight of other structures are related to the weight of the skin and stringers by empirical equations (Torenbeek, 1982; Slingerland, 2007).

The empirical method for predicting the frame weight is given:

$$W_{frame} = 0.0911(W_{skin} + W_{str})^{1.13} \tag{D.3}$$

The gross shell weight, W_{gross}, is the addition of the weight of the frames, stringers and skin.

The weight penalties are incurred by the connections with the wing, tail and engine if applicable, and cut-outs including windows, doors, emergency exits, cargo doors, undercarriage and cockpit window. The equations for calculating these penalties are given:

$$W_{window} = 6.771\sqrt{b_f} \tag{D.4}$$

$$W_{door} = doorNum(21.5\sqrt{0.000145\Delta p} + 38) \tag{D.5}$$

$$W_{escape} = escapeNum(32.2\sqrt{\frac{\Delta p}{1000g}}S_{escape} + 26.8\sqrt{S_{escape}}) \tag{D.6}$$

$$W_{cargo} = cargoNum(48.8\sqrt{\frac{\Delta p}{1000g}}S_{cargo} + 43.15\sqrt{S_{cargo}}) \quad \text{(D.7)}$$

$$W_{nosebay} = 0.0022MTOW \quad \text{(D.8)}$$

$$W_{mainbay} = 0.125(\frac{\Delta p}{1000g})^{0.8}W_{gross} \quad \text{(D.9)}$$

$$W_{cockpit} = 4.7(\frac{1.9438V_d}{100})^2\sqrt{0.000145\Delta p} \quad \text{(D.10)}$$

$$W_{wingfuse} = 20.4 + 0.000907\, n\, MTOW \quad \text{(D.11)}$$

$$W_{tailconn} = 0.1W_{tail} \quad \text{(D.12)}$$

$$W_{engfuse} = 0.03W_{enggroup} \quad \text{(D.13)}$$

where b_f and V_d are the fuselage diameter and, diving speed respectively, and Δp is the pressure differential.

The other fuselage structural parts are the front and rear pressure bulkhead, the cabin and the cargo floors. The secondary structure weight is also considered to account for the access hatches, fairings, fillets, paint, hoist and jack fitting etc. The equations are given below:

$$W_{bulk} = 9.1 + 7.225\left(\frac{\Delta p}{10000g}\right)^{0.8}(R^2 * \pi)^{1.2} \quad \text{(D.14)}$$

$$W_{flr} = 0.3074\sqrt{P_{flr}}S_{flr}^{1.045} \quad \text{(D.15)}$$

$$W_{cargoflr} = 0.3074\sqrt{P_{cargoflr}}S_{cargoflr}^{1.045} \quad \text{(D.16)}$$

$$W_{grid} = 0.66b_f^{1.35}S_{flr}^{1.045} \quad \text{(D.17)}$$

$$W_{rail} = 1.8S_{flr} \quad \text{(D.18)}$$

$$W_{sec} = 0.1W_{gross} \quad \text{(D.19)}$$

where R is the fuselage radius, P_{flr} is load intensity on the floor, and $P_{cargoflr}$ is load intensity on the cargo floor.

Appendix E Geometry of lifting surfaces of PraP300

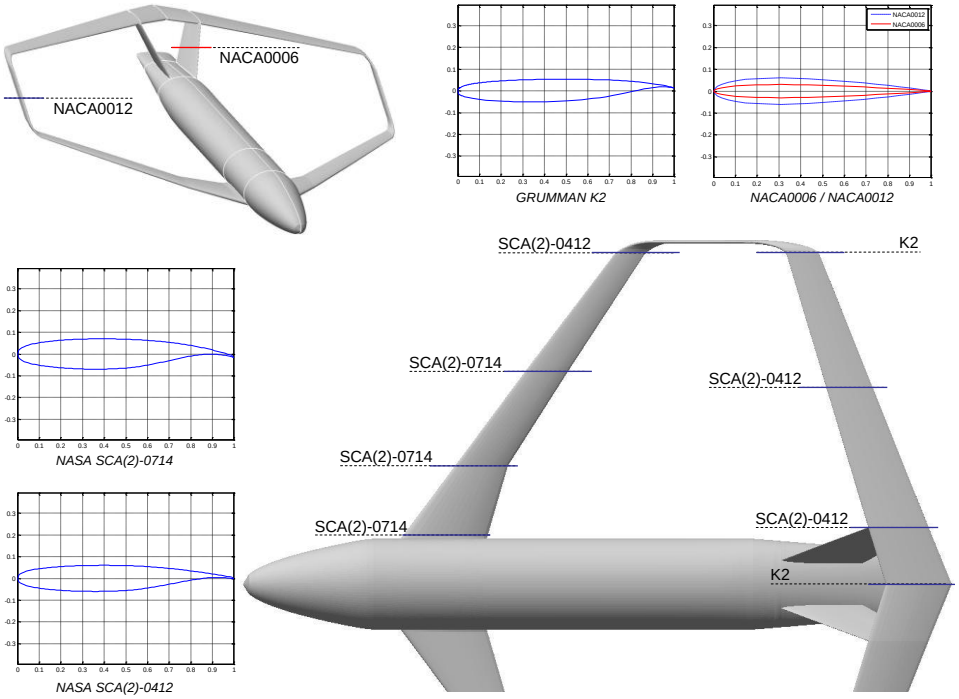
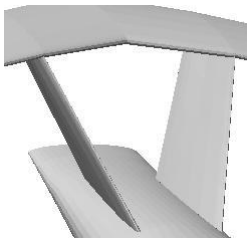


Figure E.1: PrP300 airfoil sections

Table E.1: Values for the starboard vertical fin.



	Vertical fin
Airfoil	NACA0006
x [m]	41.22793
y [m]	1.89973
z [m]	2.80000
Scale [m]	6.61923
Twist [°]	0
Airfoil	NACA0006
x [m]	47.69200
y [m]	4.30000
z [m]	7.50021
Scale [m]	4.60660
Twist [°]	0

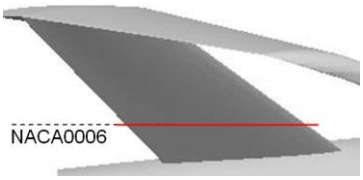
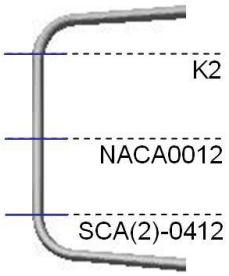


Table E.2: PrP300 wing system design

	Front wing
Airfoil	*NASA SCA(2)-0714*
x [m]	12.18204
y [m]	3.64000
z [m]	-1.32580
Scale [m]	6.47706
Twist [°]	2.00
Dihedral [°]	1.0
Taper ratio [-]	0.62
Airfoil	*NASA SCA(2)-0714*
x [m]	16.18704
y [m]	9.00000
z [m]	-1.23223
Scale [m]	4.034680
Twist [°]	1.80
Dihedral [°]	5.0
Taper ratio [-]	0.80
Airfoil	*NASA SCA(2)-0714*
x [m]	21.77704
y [m]	16.50000
z [m]	-0.57607
Scale [m]	3.23370
Twist [°]	-1.00
Dihedral [°]	5.0
Taper ratio [-]	0.72
Airfoil	NASA SCA(2)-412*
x [m]	28.27204
y [m]	25.00000
z [m]	0.16759
Scale [m]	2.32110*
Twist [°]	0
Sweep [°]	35.17
Aspect ratio	11.55

	Rear wing
Airfoil	*GRUMMAN K2*
x [m]	49.01700
y [m]	0.00000
z [m]	7.50021
Scale [m]	5.04000
Twist [°]	1.60
Dihedral [°]	0.0
Taper ratio [-]	0.91
Airfoil	*NASA SCA(2)-0412*
x [m]	47.69200
y [m]	4.30000
z [m]	7.50021
Scale [m]	4.60660
Twist [°]	1.60
Dihedral [°]	-4.0
Taper ratio [-]	0.75
Airfoil	*NASA SCA(2)-0412*
x [m]	44.20000
y [m]	15.91000
z [m]	6.68836
Scale [m]	3.43630
Twist [°]	0.55
Dihedral [°]	-4.0
Taper ratio [-]	0.73
Airfoil	GRUMMAN K2**
x [m]	41.47000
y [m]	25.00000
z [m]	6.05273
Scale [m]	2.52000**
Twist [°]	0
Sweep [°]	-18.11
Aspect ratio	13.23

	Side wing
Airfoil	*NASA SCA(2)-0412**
x [m]	30.89717
y [m]	26.00001
z [m]	1.16759
Scale [m]	2.32110*
Twist [°]	0
Airfoil	*NACA0012*
x [m]	34.51725
y [m]	25.99809
z [m]	3.11016
Scale [m]	2.42055
Twist [°]	0
Airfoil	*GRUMMAN K2***
x [m]	38.13732
y [m]	25.99618
z [m]	5.05273
Scale [m]	2.52000**
Twist [°]	0



Appendix F DoE results for the fuselage ADEE

No.	Frame pitch b_f (m)	Efficiency factor η	Weight (kg)
1	1.061	0.475	5321.0
2	0.429	0.605	4693.6
3	1.136	0.530	5181.3
4	0.733	0.412	5237.3
5	0.832	0.728	4689.7
6	1.160	0.778	4741.9
7	0.948	0.405	5538.3
8	0.464	0.896	4681.6
9	0.256	0.743	4789.4
10	0.236	0.657	4812.3
11	0.967	0.927	4633.7
12	0.688	0.698	4676.6
13	0.569	0.449	4937.4
14	0.796	0.843	4641.9
15	1.002	0.872	4649.0
16	0.620	0.550	4784.0
17	0.881	0.569	4904.0
18	0.348	0.825	4723.1
19	0.507	0.650	4676.3
20	0.370	0.363	4955.2

Appendix G Input parameters for the panel MMG

Parameter	Example	Description
Structural layout		
frame-input	‘((:frame-id 1 :material-type ‘AL7075’ :manu-method ‘machining’ :virtual? nil :orientation-center (13 0 0) :orientation 0 :section-id 1 :thickness 0.002) …)	This input parameter is a list of property lists. All the frames are defined in this parameter. :section-id indicate which are the defined section for this frame. All the sections are defined by specifying the section-input parameter.
stringer-input	‘((:stringer-id 1 :material-type ‘AL2024’ :manu-method ‘machining’ :virtual? nil :start-ref-frame-ID 1 :end-ref-frame-ID 3 :start-frame-parameter 0.3 :end-frame-parameter 0.3 :section-id 2 :thickness 0.002) …)	This input parameter is a list of property lists. All the stringers are defined in this parameter.
cutout-shape-input	‘((:cut-out-id 1 :defined-through-vertices? t :vertices-location ((5.1 0 1.2) (5.1 0 0.54) (5.5 0 0.54) (5.5 0 1.2)) : fillet-radius 0.15) (:cut-out-id 2 :defined-through-vertices?)) nil : major-radius 0.8 : minor-radius 0.6) …)	Define the shapes of cutouts. The example of this parameter shows two different types of cutout shapes: one is polygonal and the other is elliptical.
window-frame-input	‘((:window-frame-id 1 :attached-cut-out-id 1 :material-type ‘AL2024’ :manu-method ‘machining’ :section-id 2 :virtual? nil)…)	This input parameter is a list of property lists. All the cut-out flanges are defined in this parameter.

skin-layer-input	‘((:grid-coordinates (list '(0 3) '(0 9) '(20 9) '(20 3)) :sheet-id "skin-layer-1" :prop-list (:mat-name ‘AL2024’ :thickness 0.0015 :orient 45)…)	This input parameter is a list of property lists. All the skin layers are defined in this parameter.
skin-manu-method	(“cutting” “hand_layup” “consolidation”)	A list of list for defining manufacturing processes for skin.
doublers-shape-input	‘(((:grid-coordinates (list '(0 3) '(0 9) '(20 9) '(20 3)) :sheet-id "window-doubler-1" :prop-list (:mat-name ‘AL2024’ :thickness 0.0015 :orient 45)…) …)	This input parameter is a list of property lists. All the doublers are defined in this parameter. The input of each doubler is the same as skin-input.
doublers-manu-method	‘((“bonding”) (“bonding”))	A list of list for defining manufacturing processes for each doubler.
Materials & Sections		
material-input	‘((:mat-name ‘pps’ :Ex 90e09 :Ey 9.8e09 :Gxy 3.5e09 :rho 1760 :nu 0.29) (:mat-name ‘AL2024’ :E 73e9 :rho 2700 :nu 0.318) …)	This input parameter is a list of property lists. Define the mechanical properties for material.
section-input	‘(‘(:section-type :z :bar-section-name “frame” :w 0.08:t0 0.004 :h1 0.1 :h2 0.108 :offset 0.053)…)	This input parameter is a list of property lists. Define section parameters for the beam-like structures. The same as the beam-section-input for the fuselage ADEE.
global-ms-num	5	The mesh seeds number on each edge.
Load cases		
load-input	‘((:global-FEA-lc-id 1	This input parameter is a list of

	:deltaP 61500)…)	property lists. The FEA of panel will be analyzed under the first load case of global FE model.

Appendix H Input parameters for the movable MMG

Parameter	Example	Description
OML definition		
airfoils	‘("rudder-root" "rudder-tip")	The root airfoil and tip airfoil type.
root-deltac-over-c	0	0: rotates the root airfoil at the LE point. 1: rotates the root airfoil at the TE point.
tip-deltac-over-c	1	Same as above.
root-airfoil-cant-angle	0	The airfoil is streamwisely placed.
tip-airfoil-cant-angle	-30	Counterclockwisely rotate tip airfoil 30 degrees.
root-airfoil-LE-point	‘(0 0 0)	The xyz coordinates of the point.
tip-airfoil-LE-point	‘(1 2.689 0)	
root-airfoil-TE-point	‘(1.2 0 0)	
tip-airfoil-TE-point	‘(1.6 3.189 0)	
root-airfoil-thickness	200	Scaled by 200%.
tip-airfoil-thickness	100	
position	‘(:horizontal (0.0 0.0 0.0))	The OML is horizontally positioned at (0 0 0) point.
Structural members		
trunk-dividing-spar-input	‘(:root-chord-percentage 0.5 :tip-chord-percentage 0.5)	This input parameter is a list of property lists. The trunk dividing spars are used to define the trunk dividing ribs.
trunk-dividing-rib-input	‘((:rib-cut-plane-center 0.5 :angle-with-streamwise 0) …)	This input parameter is a list of property list. The trunk dividing ribs are used to divide different trunks from an OML.
trunk-input	‘((:spar-input (spar1-input spar2-input …) :WB-rib-input (rib1-input rib2-input …) :LE-rib-input (LE-rib1-input LE-rib2-input …) :TE-rib-input (TE-rib1-input TE-rib2-	Define the spars and WB/LE/TE ribs for each trunk. The input parameters for defining these parts are given in Section 6.5.

	input …)))	
hinge-line-input	‘(:offset-in-x-root 0.01 :offset-in-z-root 0.01)	Define the hinge line.
actuator-line-input	‘(:offset-in-x-root 0.01 :offset-in-z-root 0.01)	Define the actuator line.
hinge-input	‘((:hinge-id 1 :node-id 10004 :trunk-id 1 :attached-rib-id 3)…)	A list of property lists. :trunk-id indicates to which trunk the hinge is attached. The ID of the node which represents the hinge is 10005
actuator-input	‘((:actuator-id 1 :node-id 10005 :trunk-id 1 :attached-rib-id 3)…)	A list of property lists. :trunk-id indicates to which trunk the actuator is attached.
manufacturable-spar-input	‘((:manu-spar-id 1 :segments ((1 2) (2 1) (3 1))))	A list of property list for defining a manufacturable spar, using the geometry generated from the movable trunks. The contour of this manufacturable spar is composed of the contours of the 2nd spar of the 1st movable trunk, and the 1st spar of the 2nd movable.
manufacturable-skin-input	‘((:manu-skin-id 1 :segments ((1 0) (2 0) (3 0))))	Similar to the definition of a manufacturable spar. 0 means the upper skin surface of a movable trunk, whereas 1 indicates the lower surface. This example means the skin is composed of the upper skins of trunks 1-3.
Properties & mesh		
material-input	‘((:mat-name ‘pps’ :nu 0.29 :Ex 90e09 :Ey 9.8e09 :Gxy 3.5e09 :rho 1760) (:mat-name ‘AL2024’ :E 73e9 :rho 2700 :nu 0.318) …)	This input parameter is a list of property list. Define the mechanical properties for material.
single-ply-thickness	0.138e-3	The thickness of a single ply.

spar-cap-prop-input	‘(:prop-name ‘spar-cap’ :material ‘pps’ :section-parameters (:section-type :bar :w 0.01 :h 0.002))	A list of property lists which contains the property data required for assigning the properties for spar caps in PATRAN.
rib-cap-prop-input	‘(:prop-name ‘rib-cap’ :material ‘pps’ :section-parameters (:section-type :bar :w 0.01 :h 0.002))	A list of a property list which contains the property data required for assigning the properties for spar caps in PATRAN.
initial-thickness	‘(:spar-thickness 0.002 :rib-thickness 0.001 :skin-thickness 0.002)	Define the initial thickness of structural members.
global-ms-num	5	The mesh seeds number on each edge.
Cost estimation		
part-production-method-input	‘(:WB-rib ‘press_forming’ :LE-rib ‘press_forming’ :TE-rib ‘press_forming’ :spar “hand_layup” :skin “hand_layup”)	This input parameter is a list of property list which is used to define the production method for ribs, spars and skin.
cost-estimation-constants	‘(:pl 0.3 :sr 0.5 :pm 50 :nu0-rib 5 :tao0-rib 10 :nu0-spar 5 :tao0-spar 10 :nu0-skin 5 :tao0-skin 10 :nu0-assy 0.05 :tao0-assy 10)	Constants used for bottom-up parametric cost estimation.
Loads		
node-displacement	‘((:bc-node-id (list 10004) :ref-coord 1 :bc-name "fixed-actuator-1" :DOFs (list "<0,,>" "<,,>"))) …)	A list of property lists. The x direction translation of the node 10004 is constrained.
load-case-input	‘((:loadcase-id 1 :loadcase-scale 1.0 :load-names ("pressure-load-upper") :priority (list 0) :load-scale (list 0.0))…)	A list of property lists required by the PCL-writer to define load cases.

Appendix I Movable ADEE DoE results

No.	x1	x2	x3	x4	Cost ($)	Weight (kg)
1	4	15	0.282	0.713	29524	57.78
2	3	16	0.292	0.707	28113	66.28
3	3	14	0.240	0.783	27971	63.15
4	2	13	0.260	0.708	22724	67.46
5	2	14	0.233	0.732	23637	60.48
6	3	13	0.298	0.776	26051	59.47
7	4	14	0.263	0.750	30481	61.67
8	3	16	0.277	0.796	28350	62.12
9	3	16	0.236	0.755	28285	62.74
10	2	17	0.208	0.744	25182	61.00
11	3	13	0.246	0.740	26987	63.18
12	4	13	0.286	0.707	28867	59.19
13	2	17	0.268	0.766	24634	74.44
14	2	15	0.239	0.768	23932	60.08
15	4	16	0.248	0.773	31852	63.59
16	2	15	0.211	0.765	25091	64.98
17	3	13	0.270	0.727	26808	80.86
18	4	15	0.212	0.701	30784	60.53
19	4	15	0.266	0.734	30232	59.87
20	3	17	0.207	0.793	29828	64.01
21	4	14	0.259	0.774	30743	61.95
22	3	14	0.280	0.731	26939	62.58
23	4	14	0.255	0.724	30680	63.42
24	3	17	0.269	0.721	28418	64.06
25	2	16	0.238	0.711	23588	58.68
26	3	14	0.223	0.746	28208	65.11
27	3	16	0.299	0.729	27417	62.09
28	3	14	0.295	0.726	27597	67.53
29	3	15	0.275	0.716	26509	59.36
30	2	17	0.275	0.753	24393	85.80
31	3	15	0.263	0.769	27269	66.77
32	3	14	0.206	0.739	27908	62.71
33	3	16	0.249	0.718	27321	59.79
34	4	14	0.288	0.714	30334	63.97
35	3	13	0.216	0.793	27737	63.36
36	3	14	0.285	0.736	26914	62.44

No.	x1	x2	x3	x4	Cost ($)	Weight (kg)
37	3	15	0.215	0.719	28030	63.93
38	3	16	0.244	0.789	28409	62.25
39	4	15	0.224	0.780	31386	61.67
40	3	14	0.229	0.771	28260	64.97
41	3	14	0.294	0.754	26749	61.41
42	4	17	0.201	0.742	33089	64.32
43	2	15	0.272	0.787	24240	62.51
44	3	15	0.232	0.722	26984	58.73
45	4	17	0.258	0.797	32080	60.87
46	3	13	0.227	0.784	27708	64.30
47	2	16	0.290	0.763	24011	60.02
48	3	15	0.290	0.751	26920	61.01
49	2	15	0.252	0.791	24224	60.89
50	3	16	0.203	0.799	29497	64.85
51	2	15	0.250	0.703	23161	59.47
52	2	16	0.283	0.786	24288	59.69
53	4	16	0.254	0.762	30996	59.70
54	2	16	0.217	0.748	24750	61.78
55	3	15	0.226	0.757	28383	65.18
56	3	14	0.204	0.779	28406	63.67
57	2	16	0.231	0.760	24269	59.40
58	4	15	0.219	0.778	32116	65.87
59	3	17	0.221	0.705	29946	70.39
60	4	14	0.243	0.737	31099	64.40

References

ACARE, “Strategic Research Agenda Executive Summary”, Advisory Council for Aeronautics Research in Europe, 2002.

Alexandrov N., Hussaini M. Y., editors, Multidisciplinary Design Optimization: State-of-the-Art, SIAM, 1997.

Alhir S. S., “UML in a Nutshell: A Desktop Quick Reference”, O’Reilly, Beijing, 1998.

Allison, J. T., Walsh, D., Kokkolaras, M., Papalambros, P. Y., Cartmell, M., “Analytical Target Cascading in Aircraft Design,” Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 2006.

Anderson J.D., Aircraft Performance and design, McGraw-Hill, 1998.

Anumba C.J., Baldwin A.N., Bouchlaghem D., Prasad B., Cutting-Decelle A.F., Dufau J., Mommessin M., “Integrating concurrent engineering concepts in a steelwork construction project,”Concurr Eng: Res Appl, Vol. 8, No. 3, 2000,pp. 199–211

Ardema M.D., Chambers M.C., Patron A.P., Hahn A.S., Miura H., Moore M.D., Analytical fuselage and wing weight estimation of transport aircraft, NASA Technical Memorandum 110392, May 1996.

Bao H.P., Samareh J.A., Affordable design: a methodology to implement process-based manufacturing cost models into the traditional performance focused multidisciplinary design optimization. AIAA-2000-4839, 2000.

Bazilevs Y., Hsu M.C., Scott M.A. , “Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations and with application to wind turbines”, Computer Methods in Applied Mechanics and Engineering, Vol. 249, No. 252, 2012, pp. 28-41.

Berg T.van den, Parametric modelling and aerodynamic analysis of multi-element wing configurations, MSc thesis. Delft University of Technology, 2009.

Blanchard B.S., Fabrycky W.J., Systems Engineering and Analysis, Prentice Hall International Series in Industrial and Systems Engineering, Pearson Prentice Hall, Upper Saddle River, NJ, 5th edition, 2011

Brouwers Y.H.A., Development of a KBE application to support the conceptual design of passenger aircraft fuselages. , MSc thesis. Delft University of Technology, 2011.

Castagne S., Curran R., Rothwell A., Price M., Benard E., Raghunathan S., “A generic tool for cost estimating in aircraft design,” Res Eng Design, Vol. 18, No. 4, 2008, pp. 149-162.

Cerulli C., Schut E.J., Berends J.P.T.J., Tooren M.J.L.v., “Tail Optimization and Redesign in a Multi Agent Task Environment”, 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, USA, 2006.

Chapman C. B., Pinfold, M., “Design Engineering – A Need to Rethink the Solution Using Knowledge Based Engineering,” Knowledge-Based Systems, Vol. 12, 1999, pp. 257-267.

Chapman C.B., Pinfold M., "The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure," 2001, Advances in Engineering Software, Vol. 32, pp. 903-912.

Chiciudean T.G., Cooper C.A, "Design of an Integral Pre-processor for Wing-like Structure Multi-Model Generation and Analysis," Proceedings of the 27th International Congress of the Aeronautical Sciences, Nice, France, 19-24 September 2010.

Choi J.W., Kelly D., Raju J., Reidsema C., "Knowledge-Based Engineering System to Estimate Manufacturing Cost for Composite Structures," Journal of Aircraft, Vol. 42, No. 6, 2005, pp. 1396-1342.

Choi J.W., Kelly D., Raju J., "A Knowledge-Based Engineering Tool to Estimate cost and weight of Composite Aerospace Structure at the Conceptual Stage of the Design Process," Aircraft Engineering and Aerospace Technology, Vol. 79, No. 5, 2007, pp. 459-468.

Clark K.B., Fujimoto T., Product Development Performance: Strategy, Organization and Management in the World Auto Industry, Harvard Business School Press, Cambridge, 1991.

Cooper C., Development of methodology to support design of complex wings, PhD thesis. Delft University of Technology, 2011.

Curran R., Raghunathan S., Price M.,"Review of aerospace engineering cost modelling: The genetic causal approach", Progress in Aerospace Sciences, Vol. 40, 2004, pp. 287-534.

Curran R., Rothwell A., Castagne S., "Numerical Method for Cost-Weight Optimization of Stringer-Skin Panels," Journal of Aircraft, Vol. 43, No. 1, 2006, pp. 264-274.

Currey N.S., Aircraft landing gear design: principles and practices, AIAA Education Series, Washington D.C., 1996.

Dalen F. v., "Modeling a Large Fuselage with ADAS," Report B2V-98-20, Delft University of Technology, July 1998.

Dattoma V., Giorgi M.D., Giancane S., Manco P., Morabito A.E., "A parametric–associative modelling of aeronautical concepts for structural optimization," 2012, Advances in Engineering Software, Vol. 50, pp. 97–109.

Department of Defence, "Systems Engineering Fundamentals", Defence Acquisition University Press, USA, 2001

Doherty J.J., Dean S.R.H., "MDO-based concept optimisation and the impact of technology and systems choices," 7th AIAA Aviation Technology, Integration and Operations Conference, Northern Ireland, 2007.

Dorbath F., Nagel B., Gollnick V., A knowledge based approach for automated modelling of extended wing structures in preliminary aircraft design, 60. Deutsche Luft- und Raumfahrtkongres, 2011

Dorbath F., Nagel B., Gollnick V., "Implementation of a Tool Chain for Extended Physics-Based Wing Mass Estimation in Early Design Stages,"71th Annual Conference of Society of Allied Weight Engineers (SAWE), 2012.

Dorst K., Vermaas P.E., Gero J.S., "Function-Behaviour-Structure model of designing: a critical analysis, Research in Engineering Design," Vol. 16, 2005, pp. 17-26.

Drela M., Youngren H., AVL (Athena Vortex Lattice) 3.27, Massachusetts Institute of Technology, 2008.

Duverlie P., Castelain J.M., "Cost Estimation During Design Step: Parametric Method versus Case Based Reasoning Method", International Journal of Advanced Manufacturing Technology, Vol. 15, 1999, pp. 895-906.

Elham A., La Rocca G., Vos R., "Refined Preliminary Weight Estimation Tool for Airplane Wing and Tail", SAE Technical Paper 2011-01-2765, 2011.

Elham A., La Rocca G., van Tooren M.J.L.. "Development and implementation of an advanced, design-sensitive method for wing weight estimation", 2013. Aerospace Science and Technology, Vol. 29, No. 1, pp. 100-113.

Emberey C.L., Milton N.R. et.al, Application of Knowledge Engineering Methodologies to Support Engineering Design Application Development in Aerospace, 7th AIAA Aviation Technology, Integration and Operations Conference, Northern Ireland, 2007.

Engels H., Becker W., Morris A., "Implementation of a multi-level optimization methodology within the e-design of a blended wing body", 2004, Aerospace Science and Technology, vol. 8, pp. 145-153.

European Aviation Safety Agency, Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes, CS-25, July, 2011.

Frediani A., A 250 Passenger PrandtlPlane transport aircraft preliminary design. In: XVIIICongresso Nazionale AIDAA, Volterra, Italy, 19–22 September 2005.

Frediani A., The Prandtl Wing. V.K.I. Lecture series: Innovative Configurations and Advanced Concepts for Future Civil Transport Aircraft, 06–10 June 2005.

Forsberg K., Mooz H., "The relationship of systems engineering to the project cycle,"1992, Engineering Management Journal, Vol. 4, No. 3, pp. 36-43.

Gero J.S., "Design prototypes: a knowledge representation schema for design," AI Magazine, Vol. 11, No. 4, 1990, pp. 26-36.

Gero J.S. and Kannengiesser U., "The situated Function-Behaviour-Structure framework," Design Studies, Vol. 25, No. 4, 2004, pp. 373-391.

Ginneken D.van, Automatic control surface design & sizing for the Prandtl plane, MSc Thesis. Delft University of Technology, 2009.

Ginneken D.van, Voskuijl M., Tooren M.J.L.van, Frediani A., Automated control surface design and sizing for the PrandtlPlane. In: 6th AIAA Multidisciplinary Design Optimization Specialist Conference, Orlando, Florida, 12–15 April 2010.

Heinze W., Lender S., Wang H., Critical load cases for aircraft fuselage, Discussion Notes, May 3, 2011.

Heinze W., “Ein Beitrag zur Quantitativen Analyse der Technischen und Wirtschaftlichen Auslegungsgrenzen Verschiedener Flugzeugkonzepte für den Transport Grosser Nutzlasten,” Technische Univ. Braunschweig, Zentrum für Luft- und Raumfahrt, Rept. 94-01, Braunschweig, Germany, 1994.

Hinte E. van, Tooren M.J.L. van, First read this, 010 publishers, the Netherlands, 2008

Hoek M.van, Structural design, analysis and optimization of a lifting surface in a knowledge based engineering environment, Mater thesis, Delft, 2010.

Hurlimann F., Kel R., Dugas M., Oltmann K., Kress G., “Mass estimation of transport aircraft wingbox structures with a CAD/CAE-based multidisciplinary process,” Aerospace Science and Technology, Vol. 15, No.4, 2011, pp. 323-33.

IDA Report R-338. Alexandria. USA: Institute for Defense Analyses (IDA), 1986.

Ilcewicz L.B., Mabson G.E., Metschan S.L., Swanson G.D., Proctor M.R., Tervo D.K., Fredrikson H.G., Gutowski T.G., Neoh E.T., Polgar K.C., “Cost Optimization Software for Transport Aircraft Design Evaluation (COSTADE), Design Cost Methods”, 1996, NASA Contractor Report 4737.

INCOSE, Systems Engineering Handbook; A Guide for System Lifecycle Processes and Activities, International Council on Systems Engineering (INCOSE), 3rd edition, 2006.

International Cost Estimation and Analysis Association. “Parametric Estimating Handbook”, 4th ed., 2008, https://www.iceaaonline.org/documentation/files/ISPA_PEH_4th_ed_Final.pdf [cited 01-03-2013].

Kassapoglou C., “Simultaneous Cost and Weight Minimization of Composite Stiffened Panels under Compression and Shear,” Composite Part A: Applied Science and Manufacturing, Vol. 28, No. 5, 1997, pp. 419-435.

Kassapoglou C., “Minimum Cost and Weight Design of Fuselage Frames. Part a: Design Constraints and Manufacturing Process Characteristics,” Composite Part A: Applied Science and Manufacturing, Vol. 30, No. 7, 1999, pp. 887-894.

Kassapoglou C., “Minimum Cost and Weight Design of Fuselage Frames. Part b: Cost Considerations, Optimization, and Results,” Composite Part A: Applied Science and Manufacturing, Vol. 30, No. 7, 1999, pp. 895-904.

Kassapoglou C., Design and analysis of composite structures: with applications to aerospace structures, AIAA education series, Reston, 2010.

Kaufmann M., Zenkert D., Wennhage P., “Integrated Cost/Weight Optimization of Aircraft Structures,” Structural and Multidisciplinary Optimization, Vol. 41, No. 2, 2010, pp. 325–334.

Kollár L. P., Springer, G. S., Mechanic of Composite Structures, Cambridge University Press, Cambridge, 2003.

Krammer J., Sensburg O., Vilsmeier J., Berchtold G., Concurrent Engineering in Design of Aircraft Structures, Journal of Aircraft Design, Vol. 32, No. 2, 2002.

Krakers L.A., Parametric fuselage design - Integration of mechanics and acoustic & thermal insulation, PhD thesis. Delft University of Technology, 2009.

Krishnan V., Ulrich K.T. "Product development decisions: A review of the literature," Management Sci., Vol. 47, No. 1, 2001, pp. 1–21.

Kroo I. M., "MDO for Large-Scale Design," Multidisciplinary Design Optimization: State-of-the-Art, edited by N. Alexandrov and M. Y. Hussaini, SIAM, 1997, pp. 22–44.

Laan A.H. van der, Tooren M.J.L. van, "Parametric modeling of movables for structural analysis", Journal of aircraft, Vol. 42, No. 6, 2005, pp. 1606-1614.

Laan A.H. van der, Knowledge based engineering support for aircraft component design, PhD thesis. Delft University of Technology, 2008.

La Rocca G., Krakers L.A., Tooren M.J.L.van, "Development of an ICAD generative model for blended wing body aircraft design", 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, USA, 2002.

La Rocca G., Tooren M.J.L.van, "A knowledge based engineering approach to support automatic generation of FE models in aircraft design," 9 45th AIAA Aerospace Sciences Meeting, USA, 2007.

La Rocca G., Knowledge based engineering techniques to support aircraft design and optimization, PhD thesis. Delft University of Technology, 2010.

La Rocca G., Tooren M.J.L.van, "Knowledge-based engineering to support aircraft multidisciplinary design and optimization," Proceedings of the Institution of Mechanical Engineering, Part G: Journal of Aerospace Engineering, 2010.

La Rocca G., "Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design," Advanced Engineering Informatics, Vol. 26, Issue 2, 2012, pp. 159–179.

Leung ACK, Wainwright CER, Leonard R., "The development of an integrated cost estimation system," Int J Comput Integrated Manuf, Vol. 9, Issue 3, 1996, pp 190–204.

Liker J.K., Sobek, D.K., Ward, A.C., Cristiano, J.J., " Involving suppliers in product development in the United States and Japan: Evidence for set-based concurrent engineering," IEEE Transactions on Engineering Management, Vol. 43, No. 2, 1996, pp. 165–178.

Liker J.K., Kamath R.R., Wasti S.N., and Nagamachi M., "Integrating Suppliers into Fast-Cycle Product development," Engineered in Japan: Japanese Technology Management Practices. Oxford University Press, Oxford, 1995.

Locatelli D., Mulani S.B., and Kapania R.K., "Wing-box weight optimization using curvilinear spars and ribs (SpaRibs)," Journal of Aircraft, Vol. 48, No. 5, 2011, pp. 1671-1684.

Locatelli D., Mulani S.B., Kapania R.K., "Parameterization of curvilinear spars and ribs (SpaRibs) for Optimum Wing Structural Design," 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, Hawaii, AIAA 2012-1359, 2012.

Lomax T.L., Structural loads analysis for commercial transport aircraft: Theory and Practice. AIAA Education Series, Washington D.C., 1996.

Lophaven S., Nielsen H.B., Sondergaard J., “DACE, a MATLAB Kriging toolbox,” Technical report IMM-TR-2002-12, Information and mathematical modeling institute, Technical University of Denmark, August 2002.

Mathworks, MATLAB optimization toolbox - R2012b, 2012.

MSC, MSC Software, MacNeal-Schwendler Corporation, 2014, available at www.mscsoftware.com.

Mawhinney P., Price M., Curran R., Benard E., Murphy A., Raghunathan, S., “Geometry-based approach to analysis integration for aircraft conceptual design”, Sep. 2005, AIAA-2005-7481, 5th Annual Aviation Technology, Integration & Operations (ATIO) Forum, Washington, DC, USA.

Mavris D.N., Pinon O.J., A Systems Engineering Approach to Aircraft Design, Encyclopedia of Aerospace Engineering, John Wiley, 2012.

Miller J.H. and Page S.E., Complex adaptive systems: an introduction to computational models of social life. Princeton University Press, Princeton, 2007.

Milton N., Knowledge Acquisition in Practice. Springer-Verlag, 2007.

Mitchell M., Complexity: a guided tour. Oxford University Press, Oxford, 2009.

Minnet S. and Talyor, C., Building the World’s Largest Passenger Aircraft Wings. http://www.cadinfo.net/editorial/A380-1. Date accessed: 6 May 2008.

Nawijn M., Tooren M.J.L.van, Berends J.P.T.J., “Automated Finite Element Analysis in a Knowledge Based Engineering Environment”, 2006, 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, USA.

Nevins J. L. and Whitney D. E. et al., Concurrent Design of Products and Process, McCraw-Hill, New York, 1989.

Niu M. C. Y., Airframe Structural Design. Conmilit Press LTD, Hong Kong, 1988.

Niu M. C. Y., Airframe Stress Analysis and Sizing, Conmilit Press Ltd., Hong Kong, 1997.

Pardessus T., “Concurrent engineering development and practices for aircraft design at airbus,” 24th International Congress of The Aeronautical Sciences, Yokohama, Japan, 2004.

Piegl L. and Tiller W. (1997). The NURBS Book; Monographs in Visual Communication. Springer, Berlin, 2nd edition.

Prandtl L., Induced drag of multiplanes. Technische Berichte, III(7), 1924.

Price M., Raghunathan S.; Curran R., “An integrated systems engineering approach to aircraft design,” Progress in Aerospace Sciences, Vol. 42, 2006, pp. 331-376.

Raymer D.P. Aircraft design: A conceptual approach. 4th ed., Washington D.C., 2006.

Rothwell A., “Structural Optimisation,” Lecture Notes, Faculty of Aerospace Engineering, Delft Univ. of Technology, Delft, The Netherlands, 2006.

Roskam J., Airplane Design Part V: Component Weight Estimation, DARcorp, Lawrence, KS, 2003.

Schrage D., Beltracchi, T., Berke, L., Dodd, A., Niedling, L. and Sobieski, J., "Current state of the art on multidisciplinary design optimization (MDO)," AIAA Technical report, 1991.

Schut J., Conceptual design automation, PhD thesis. Delft University of Technology, 2010.

Slingerland R., Alkemade F., Vermeulen B., "A preliminary prediction method for the effect of new fuselage materials on transport aircraft weight," 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, USA, 2007.

Simposn T.W., Martins J., "Multidisciplinary Design Optimization for Complex Engineered Systems: Report from a National Science Foundation Workshop," ASME Journal of Mechanical Design, Vol. 133, Issue 10, pp. 101002.

SMLib, Solid Modeling Solution, 2014, available at http://www.smlib.com/.

Sobieszczanski-Sobieski J. and Haftka R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," Structural Optimization, Vol. 14, No. 1, 1997, pp. 1–23.

Soderberg L. G., Facing up to the Engineering gap, McKinsey Quart, Vol. 25, 1989, pp. 3–23.

Stokes M., Managing Engineering Knowledge–MOKA: Methodology for Knowledge Based Engineering Applications. Professional Engineering, London, 2001.

Suh, N., Complexity; theory and applications, Oxford University Press, USA, 2005.

Susman G., Ed., Integrating Design and Manufacturing for Competitive Advantage. Columbia University Press, New York, 1992.

Sussman, J., "Ideas on complexity in systems; twenty views", Massachusetts Institute of Technology, Internet resource, http://web.mit.edu/esd.83/www/ notebook/20ViewsComplexity.PDF [last visited 03-06-2013]

Takeuchi H., Nonaka I. "The New Product Development Game." Harvard Business Review, 1986, pp. 137-146.

Tang C.S., Zimmerman J, "Managing New Product Development and Supply Chain Risks: The Boeing 787 Case," Supply Chain Forum: An International Journal, Vol. 10, 2009, pp. 74-86.

Tam W.F., "Improvement Opportunities for Aerospace Design Process," Aerospace Conference and Exhibit, San Diego, California, 2004.

Terwiesch C., Loch C.H., Meyer A. De, "Exchanging preliminary information in concurrent engineering: Alternative coordination strategies," Organ. Sci., Vol. 13, No. 4, 2002, pp. 402–419.

Tomiyama T., Kiriyama T., Hidaeki T., Xue D., Yoshikawa H., "Metamodel: a Key to Intelligent CAD Systems", Research in Engineering Design, Vol. 1, 1989, pp. 19-34.

Tooren M. J. L. van, Sustainable Knowledge Growth. TU Delft Inaugural Speech Faculty of Aerospace Engineering ed. Delft: Faculty of Aerospace Engineering, 2003.

Tooren M.J.L. van, Schut E.J., Berends J.P.T.J. "Design feasilisation using knowledge based engineering and optimization techniques," 44th AIAA Aerospace Sciences meeting and Exhibit, USA, 2006.

Tooren M.J.L. van, La Rocca G., Krakers L., Beukers A., Parametric Modelling of Complex Structure Systems Including Active Components, 9th AIAA/ISSMO Symposium on MDO, USA, 2006.

Torenbeek E., Synthesis of subsonic airplane design. Springer, Delft, 1982.

Torenbeek E., Advanced Aircraft Design: Conceptual Design, Analysis and Optimization of Subsonic Civil Airplanes. John Wiley & Sons, Ltd, Oxford, 2013.

Tosserams S., Etman L. F. P., and Rooda J. E., "A Micro-Accelerometer MDO Benchmark Problem," Structural and Multidisciplinary Optimization, Vol. 41, No. 2, 2010, pp. 255–275.

Townsend J.C., Samareh J.A., Weston R. P., Zorumski W. E., "Integration of a CAD System Into an MDO Framework," NASA Technical Report, No. NASA/TM-1998-207672.

Umeda Y., Takeda H., Tomiyama T., Yoshikawa H., "Function, Behavior, and Structure". In AIENG '90 Applications of AI in Engineering, Southernpton and Berlin: Computational Mechanics Publications and Springer-Verlag, 1990, pp.177-193.

Vandenbrande J.H., Grandine T.A., Hogan T., The search of the perfect body: Shape control for multidisciplinary design optimization, 2006, 44th AIAA Aerospace Science Meeting and Exhibit, Reno, USA.

Vanderplaats G. N., "An Efficient Feasible Directions Algorithm for Design Synthesis," AIAA Journal, Vol. 22, No. 11, 1984, pp. 1633–1640.

Velden C.A. van der, Application of knowledge based engineering to intelligent design system, PhD thesis. RMIT University, 2008.

Vermeulen B., Knowledge based method for solving complexity in design problems, PhD thesis. Delft University of Technology, 2007.

Wang K., Kelly D., Dutton S., "Multi-objective optimisation of composite aerospace structures," Composite Structures, Vol. 57, Issues 1-4, 2002, pp. 141-148.

Wang W., GUO S., YANG W., "Simultaneous partial topology and size optimization of a wing structure using ant colony and gradient based methods," Engineering Optimization, Vol. 43, No. 4, 2011, pp. 433-446.

Werner-Westphal C., Heinze W., Horst P., "Structural sizing for an unconventional, environment-friendly aircraft configuration within integrated conceptual design," Aerospace Science and Technology, Vol. 12, Issue 8, 2008, pp. 184-194.

Wheelwright S.C, Clark K.B., Creating project plans to focus product development, Harvard Business Review, Vol. 70, No. 2, 1992, pp. 70–82

Yassine A., Braha D, Complex concurrent engineering and the design structure matrix method . Concurrent Engineering: Research and Applications, Sage Publications, Vol. 11, No. 3, 2003, pp. 165-176.

Yin H., and Yu X., "Integration of Manufacturing Cost into Structural Optimization of Composite Wings," Chinese Journal of Aeronautics, Vol. 23, No. 6, 2010, pp. 670-676.

Zhao X., Wang H., Curran R., Tooren, M.J.L.van, "Concurrent aerospace thermoplastic stiffened panel conceptual design and cost estimation using knowledge based engineering," Proceedings of 19th ISPE International Conference on Concurrent Engineering, Germany, 2012.

Samenvatting

Het uitbesteden dat in de laatste decennia in de luchtvaartindustrie heeft plaatsgevonden heeft een wereldwijde leveringsketen van en naar een beperkt aantal original equipment manufacturers (OEMs) opgeleverd. Door het verplaatsen van het ontwerpen van de structurele subsystemen van het vliegtuig naar toeleveranciers heeft dit geresulteerd tot multi-niveau ontwerpen. In toenemende mate focussen OEMs zich op het toekennen van eisen en het definiëren en verifiëren van subsystemen op een globaal niveau, terwijl leveranciers zich focussen op het realiseren en verbeteren van vliegtuig subsystemen op lokaal detailniveau.

Het vertrouwen op een leveringsketen voor innovatieve ontwerpen kan er voor zorgen dat OEMs te weinig gedetailleerde kennis over het ontwerpen van subsystemen verkrijgen, met name over innovatieve detailontwerpen, bijvoorbeeld composieten en nieuwe productiemethodes. Echter, in de algehele conceptuele ontwerpfase is de analyse en evaluatie van verschillende subsysteemontwerpen, door de OEM zelf, erg afhankelijk van aannames en schattingen die normaalgesproken gebaseerd zijn op statistische / empirische data. Alhoewel globale ontwerpen snel kunnen worden geanalyseerd met gebruik van aannames en schattingen, is er een kans dat kostbare ontwerpwijzingen nodig zijn als de aannames en schattingen achteraf, tijdens de latere vliegtuigontwerpfases, foutief blijken te zijn. Leveranciers die kennis op detailniveau bezitten zouden in het begin van de algehele conceptuele ontwerpfases moeten worden betrokken, door het maken van verschillende detailontwerpen en het uitvoeren van nauwkeurigere analyses en evaluaties van deze ontwerpen. Vroege detailontwerpstudies kunnen leveranciers helpen om OEMs te helpen het risico van ontwerpwijzingen door foutieve aannames en schattingen te verminderen, en om OEMs te overtuigen van nieuwe materialen en nieuwe productiemethodes.

Het doel van dit onderzoek was het ontwikkelen van een ontwerpaanpak die **leveranciers** kan ondersteunen in het snel uitvoeren van lokale detailontwerpen, waaruit kritieke resultaten zoals kosten en gewicht kunnen worden gegenereerd tijdens de algehele conceptuele ontwerpfase. Het vlug kunnen ontwerpen van een structureel subsysteem is zeer gunstig voor leveranciers die hun concurrentievermogen willen vergroten, door snel te kunnen reageren en flexibel te zijn tijdens de conceptuele ontwerpfase van het vliegtuig. Het is ook gunstig voor de OEMs om het risico op ontwerpwijzigingen door incorrecte aannames en schattingen te verminderen.

In dit onderzoek zijn meerdere problemen in het huidige ontwerpproces geïdentificeerd die een snelle studie naar structurele subsystemen van een vliegtuig belemmeren, van welke sommige moeten worden aangepakt vanaf de kant van de leverancier.

1) De afhankelijkheid van leveranciers van OEMs om coherente, consistente en vroegtijdige ontwerp informatie te ontvangen, bijvoorbeeld geometrie en belastingen, die nodig zijn om het detailontwerp te starten. Deze afhankelijkheid zorgt ervoor dat leveranciers wachten totdat alle benodigde informatie beschikbaar is van de OEMs in de voorontwerpfase van het gehele vliegtuig. Daardoor kunnen leveranciers niet pro-actief deelnemen in de conceptuele ontwerpfase van het vliegtuig, waarin het ontwerp van de structurele subsystemen sterk afhankelijk is van aannames en inschattingen.

2) Het handmatige proces dat door leveranciers wordt gebruikt om computer-ondersteund ontwerp (CAD) en analyse modellen te updaten om ontwerpwijzigingen op het globaal en detailniveau te volgen. In de conceptuele ontwerpfase van het vliegtuig zijn zowel het globale

als detailontwerp nog niet vastgelegd en vinden er vaak wijzigingen plaats. Het handmatig updaten van modellen op detailniveau vereist significante technische inzet, en vertraagd daardoor de reactie van de leverancier op wijzingen in het globale ontwerp.

3) Er is een tekort aan multidisciplinaire design optimalisatie (MDO) bekwaamheid en capaciteit op het detailontwerpniveau door een gebrek aan kennis omtrent MDO en een gebrek aan tools om parametrische product- en procesmodellen te maken. Daardoor leveren leveranciers in de korte conceptuele ontwerpfase vaak slechte (enkele) haalbare ontwerpoplossingen, in plaats van een reeks van Pareto ontwerpoplossingen.

Om deze punten aan te pakken, en dus het concurrentievermogen van de leveranciers te verhogen, wordt een aanpak met een **globale-lokale kenniskoppeling** voorgesteld, welke is opgebouwd uit twee modules op globaal en detailontwerpniveau. De module op globaal ontwerpniveau is een cross-over die het globale ontwerp vervangt en de vereiste input voor het detailontwerp levert. Deze cross-over wordt gebruikt om in de vroege ontwerpfase de globale en het lokale detailontwerp samenhangend te maken. De module op het detailontwerpniveau is een set van parametrische product- en procesmodellen van structurele vliegtuigsubsystemen die worden gebruikt om repetitieve ontwerpstappen op detailontwerpniveau te automatiseren, zodat de analyse en evaluatie van het ontwerp van subsystemen snel kan worden uitgevoerd. Om twee hoofdredenen is knowledge based engineering (KBE) toegepast om de twee modules te implementeren: 1) parameterisering van productmodellen die gebruikt kunnen worden voor automatische (her)generatie van modellen; 2) automatisering van het voorbereiden van input voor discipline-specifieke analyse-tools. Als technische implementatie voor de voorgestelde aanpak voor het vinden van een optimaal ontwerp van een complex structureel subsysteem van het vliegtuig wordt multidisciplinaire design optimalisatie gebruikt.

Er zijn drie demonstratiesystemen ontwikkeld, elk vormgegeven als een ontwerpframework genaamd de Airframe Design and Engineering Engine (ADEE), wat een gespecialiseerde **Design and Engineering Engine (DEE)** is. De Design and Engineering Engine (Tooren, 2003) is een MDO systeem gericht op het ondersteunen en versnellen van het ontwerpproces van complexe producten, doormiddel van het automatiseren van non-creatieve en repetitieve ontwerpstappen. De verifiërende ontwerpsystemen zijn de ADEE voor de vliegtuigromp, de ADEE voor romppanelen, en de ADEE voor beweegbare structurele subsystemen.

Een van de belangrijkste bijdragen van dit onderzoek is de identificatie van de problemen in het ontwerpproces van structurele subsystemen waarbij de OEM en leveranciers betrokken zijn, en hoe deze problemen kunnen worden opgelost, zodat er snelle detailontwerpen tijdens de conceptuele vliegtuigontwerpfase kunnen worden gecreëerd. Een andere bijdrage is de ontwikkeling van de globale-lokale kenniskoppeling aanpak en de demonstratiesystemen van deze nieuwe methode, die tools en methodes levert om deze problemen aan te pakken. Elke verificatietool is een ADEE, welke wordt ondersteund door KBE om het globale en detailniveau-ontwerp automatisch uit te voeren, zodat de cross-over vlug de vereiste input voor het detailniveau ontwerp kan genereren, en de detailontwerp module snel meerdere ontwerpvarianten kan genereren en analyseren.

De vliegtuigromp ADEE is gebruikt om probleem 1 aan te pakken, door de ontwerp-onafhankelijkheid van fabrikanten van romppanelen te vergroten. De vliegtuigromp ADEE is

geïmplementeerd als een cross-over, waarvoor een gewichtsschatting gebaseerd op een eindige-elementen analyse (FEA) is ontwikkeld om de effecten van het materiaal en de structurele lay-out op het gewicht van de romp te kunnen bepalen. De globale kennis is vastgelegd in de cross-over, inclusief kennis over hoe de outer mould line (OML) te generen en kennis over hoe discipline-specificieke analyses zoals belastingsberekeningen en structurele analyse met behulp van FEA uit te voeren. De ADEE is gevalideerd aan de hand van data van rompen van conventionele vliegtuigen zoals de ATR 42, Fokker 100, Boeing 737-200, Airbus A320-200 en Airbus A300B2. De ADEE voor vliegtuigrompen is ook gebruikt om een het gewicht van een vliegtuig met gekoppelde vleugels te bepalen.

De ADEE voor panelen voor een vliegtuigromp is gebruikt om probleem 2 aan te pakken, door het automatiseren van repetitieve (her)generatie van modellen die gebruikt worden tijdens het lokaal detailontwerp. De ADEE voor panelen is de detailontwerpmodule in de globale-lokale kenniskoppeling, die bestaat uit een parametrische product-module voor panelen en discipline-specifieke analyse modellen, dat wil zeggen modellen voor structurele analyse, kostschattingen en voor de evaluatie van het gewicht. De ADEE voor de romp is een cross-over die input levert voor de ADEE voor panelen. Een parametrisch productmodel voor panelen ondersteund door KBE is in de ADEE voor panelen geïmplementeerd om op flexibele wijze verschillende configuraties van romppanelen te kunnen modelleren. Deze panelen zijn opgebouwd uit meerdere lagen en onderliggende structurele elementen, zoals dwars- en langsverstijvers. Deze structurele elementen worden gemodelleerd aan de hand van de OML die gegenereerd is door de romp ADEE. De structurele analyse maakt gebruik van globale-lokale FEA waarin een globaal eindige-elementen model vanuit de cross-over wordt gebruikt om het algehele gedrag van de romp voorspellen, en een verfijnd eindige-elementen model is gecreëerd voor het onderzoeken van het gedrag van de panelen. De proceskennis met betrekking tot de panelen is in de ADEE verwerkt om het modelleren, de structurele analyse, de parametrische bottom-up kostenberekening en de gewichtsevaluatie van de panelen te kunnen automatiseren. Door gebruik te maken van het versnelde detail-ontwerpproces van de panelen kan het detailontwerp voor de panelen snel reageren op wijzingen in het globale ontwerp, waarbij consistentie in de modellen op het globale en het detailniveau kan worden gegarandeerd.

De ADEE voor bewegende structurele subsystemen is gebruikt om probleem 3 aan te pakken, door repetitieve ontwerpstappen in het MDO proces te automatiseren. De ADEE voor bewegende structurele subsystemen, bijvoorbeeld richtings- en hoogteroeren, is ontwikkeld om multi-objective kosten-/gewichtsoptimalisaties uit te voeren, inclusief grote topologische variaties van de structurele configuratie. De door KBE ondersteunde modelleermodule van deze ADEE is in staat om, geheel geautomatiseerd, zeer uiteenlopende productconfiguraties en varianten te modelleren, en hieruit alle data die nodig is voor de gewichts- en kostenschattingsmodules te verkrijgen. De methode voor de gewichtsschatting maakt gebruik van FEA om de interne spanningen in de structurele elementen te bepalen, en een analytische dimensioneringsmethode voor composieten platen om hun minimaal benodigde dikte te bepalen. De ontwikkeling van de module om de fabricagekosten te schatten is gebaseerd op een kostenmodel uit de literatuur. De mogelijkheden van het framework zijn met succes gedemonstreerd door het ontwerpen en optimaliseren van de composieten structuur van het

richtingsroer van een zakenvliegtuig. De casus laat zien dat deze ADEE in staat is om snel de Pareto-optimale set voor minimaal structureel gewicht en fabricagekosten te vinden.

De ontwikkelde demonstratiesystemen laten zien dat een aanpak met globale-lokale kenniskoppeling leveranciers kan ondersteunen in hun wens om snel structurele subsystemen te ontwerpen gedurende de conceptuele ontwerpfase van het vliegtuig.

About the Author

Haiqiang Wang

Born 16 October 1987 in Lu'an, China

2001-2004	Pre-university education at Lu'an First High School
2004-2008	Mechanical Engineering at Nanjing University of Aeronautics and Astronautics (NUAA). Received Outstanding Student Leader Award once and Outstanding Student Scholarship four times.
2008-2009	Graduate study at Manufacturing Engineering of Aeronautics and Astronautics at NUAA, under supervision of Prof. Liao Wenhe.
2009-2014	PhD researcher at Delft University of Technology, under supervision of Prof.dr.ir M.J.L van Tooren and Dr. Gianfranco La Rocca. The research into Airframe Design and Engineering Engine improves the airframe structural design process.